AD-A229 216

DTIC FILE COPY

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

ELECTE OCT 0 9 1990

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED
4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A	5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A
6a. NAME OF PERFORMING ORGANIZATION UNIVERSITY OF CENTRAL FLORIDA 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION US ARMY PROJECT MANAGER FOR TRAINING DEVICES
6c. ADDRESS (City, State, and ZIP Code) University of Central Florida Division of Sponsored Research PO Box 25000 Orlando, FL 32816-0001	7c. ADDRESS (City, State, and ZIP Code) PM TRADE ATTN: AMCPM-TND-ET 12350 Research Parkway Orlando, FL 32826-3276
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N61339-90-C-0031
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62727 / PROJECT NO. A230 / TASK NO. A230C4 / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
(U) Performance of Distributed Simulation Using Multiple Segment Ethernet Communication Networks

12. PERSONAL AUTHOR(S)
DR. R. GUHA, DR. M. BASSIOUNI

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
final	FROM Feb 90 TO Aug 90	90-9-21	56

16. SUPPLEMENTARY NOTATION
A companion report is titled "Execution Time Requirements of Petri net Programs in a Sun Workstation Environment"

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Petri nets, Distributed Simulation, Ethernet, Simulation Networking, SIMNET, Simulation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper presents the work with the generalized stochastic Petri net (GSPN models of a simulation network similar to SIMNET, a distributed system of training simulators connected over an Ethernet local area network. In contrast to isolation models, these models use a global approach to represent the details of the interaction among the stations over the network. In our first GSPN model, sixteen chains are used to capture the state of transmission for the sixteen attempts allowed for each node. The "frame transmitter process" is accurately represented in this model by keeping track of the individual attempts separately and using the correct backoff delay distribution for each attempt. Two modifications of this model are obtained by introducing approximations that allow the execution of networks with a larger number of nodes while maintaining a good degree of accuracy. Results of important performance measures (e.g., frame loss rate due to excessive collision count) obtained from the three models are presented and compared.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL MARY C. FISCHER	22b. TELEPHONE (Include Area Code) 407-380/4348/8175 22c. OFFICE SYMBOL AMCPM-TND-ET

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Contractor & Address:	University of Central Florida Division of Sponsored Research P.O. Box 25000 Orlando, Fl 32816-0001
Contract Number:	N61339-90-C-0031
Project Title:	Petri Net Models for Evaluating the Performance of Distributed Simulation Using Several Network Architectures
Principal Investigators:	Ratan Guha and M. Bassiouni
Telephone No.	407-275-2341
Title of Report:	Phase I Technical Reports (Final Version) Performance of Distributed Simulation Using Multiple Segment Ethernet Communication Networks
CDRL No:	A003 [2]
Period of Report:	February 2, 1990 to August 7, 1990
Date of Report:	September 21, 1990
Security Classification:	Unclassified
Name of Issuing Government Activity:	DOD/NTSC (ARI/PM-TRADE)
COTR Name and Address:	Dr. Mary C. Fischer AMCPM-TND-ET PM-Trade 12350 Research Parkway Orlando, FL 32826-3276
Contracting Officer: **Attention Code:**	Mr. Rex Branch 631
Address:	Naval Training Systems Center 12350 Research Parkway Orlando, FL 32826-3275

Performance of Distributed Simulation Using Multiple Segment Ethernet Communication Networks

ABSTRACT

We are presenting results of our work with generalized stochastic Petri net (GSPN) models of a simulation network similar to SIMNET, a distributed system of training simulators connected over an Ethernet local area network. In contrast to isolation models, these models use a global approach to represent the details of the interaction among the stations over the network. In our first GSPN model, sixteen chains are used to capture the state of transmission for the sixteen attempts allowed for each node. The "frame transmitter process" is accurately represented in this model by keeping track of the individual attempts separately and using the correct backoff delay distribution for each attempt. Two modifications of this model are obtained by introducing approximations that allow the execution of networks with a larger number of nodes while maintaining a good degree of accuracy. Results of important performance measures (e.g., frame loss rate due to excessive collision count) obtained from the three models are presented and compared.

INTRODUCTION

The performance of the Ethernet local area communication network has been extensively analyzed with varying degrees of detail ([Go'85], [Gr'85],[MaCh'87], [ToHu'80],[TaKl'85]). In this paper, we describe a Stochastic Petri Net model of a specific distributed application using Ethernet. The distributed application we consider is provided by a network of training simulators which interact with each other using communication over a local area network. Although the network has been analyzed using two different models of Ethernet, only one will be described. The approach used in this model is to simulate the concurrent behavior of all stations rather than modeling only one station along with the expected effects of the remaining stations on this one station, as done by Greisser [Gr'85]. In this regard, our model is similar to that of Marsan et al. [MaCh'87]. Our model, however, differs from Marsan's model in that we model the message loss which occurs when the number of allowable transmission attempts is exceeded. We are primarily interested in the loss of messages since this can have both short- and long-term effects on the appearance of the simulation to the trainees. Several models are described in this paper. The first one gives an exact representation of the collision handling process in Ethernet. The other models

A-1

are based on close approximations of this process to allow solutions with a larger number of nodes. In the following section, we describe the operation of the simulation network then proceed to present the Stochastic Petri Net models of the combined training simulator/Ethernet combination. Our presentation begins with a description of the most detailed model. Following this, simplifications are made to develop models allowing the solution of networks with larger number of nodes. A description of the performance measures follows. Throughout the paper, the words "station" and "node" are used interchangeably.

SIMNET

SIMNET is an experimental project [Po '89],[FrHa '88] sponsored by DARPA and the US Army (PMTRADE) and developed by researchers at BBN to develop a distributed battlefield training simulation system by networking large numbers of interactive combat vehicle simulators and their supporting elements. A detailed description of the SIMNET system is given in [Po '89], [FrHa '88]. Since our Petri net model development is influenced by the SIMNET system, A brief description of this system and its networking scheme follows. Excerpts from [Po'89], [FrHa'88] provide this description.

The SIMNET vehicle simulators and their supporting elements communicate via local area and long haul networks. Simulators at a single site are connected via a 10 Megabit per second Ethernet (Ethernet is a registered trademark of the Xerox Corporation). Each Ethernet is connected to a single long haul network by a gateway. Since our model development at this point is for a single site, we restrict the discussion only to local area networks.

At each SIMNET site there is a set of tank and aircraft simulators. The Management, Command and Control (MCC) system at the site initializes these simulators and simulates indirect fire, close air support, and resupply and repair vehicles. Each simulator periodically broadcasts an update of its status (position, orientation, velocity, and other descriptive characteristics). These update messages are the major source of traffic on the network. A dead reckoning algorithm is used to decrease the number of these messages. In this algorithm, each simulator extrapolates the status of other simulators . Each simulator also extrapolates its own status. If this extrapolated information differs from the actual information by more than a predetermined amount, the simulator transmits a new update message so that it can be used by all simulators on the network. These update messages are transmitted according to a protocol, called the simulator protocol, which has been implemented as an application layer protocol making direct use of the network layer services with

no intermediate layers. In addition there are other protocols which implement the other functions of SIMNET. However, the impact of the simulator protocol on the network traffic is the most significant one (compared to the other protocols).

THE MODELING APPROACH

It is evident that there are two distinct architectures that need to be modeled and combined together to make the system development modular. One is the architectures of the communication network and the other is the distributed SIMNET architecture which utilizes the communication protocol for information transfer. Since Ethernet is available commercially, the communication architecture at a single site has been implemented by Ethernet and all SIMNET protocols are implemented as application layer protocols on top of Ethernet. In our modeling efforts, we first develop a high level model for each of the two architectures and then gradually develop the low level details by expanding on the high level models. Even at this high level of modeling, the development of the details of the communication network architecture is necessary to determine any data loss due to communication networks. We start by developing a global model of the network architecture with a very high level description of the SIMNET architecture and details of Ethernet frame transmission attempts. We then modify this global model of the network architecture by introducing approximations that allow the execution of a larger number of nodes while maintaining a high degree of accuracy.

GSPN MODEL OF SINGLE SITE SIMULATION NETWORKS WITH ETHERNET

The transmission process of a given node consists of three parts corresponding to the three processes defined by the Ethernet protocol or the MAC sublayer of IEEE standard 802.3 [Ie '85]. These processes are the frame transmitter process, the deference process, and the bit transmitter process .

Global Model 1:

This global model incorporates strong interaction among all stations in the simulator network in order to capture the semantics of operation of the network protocol accurately. To obtain an analytically solvable model based on the underlying Markov chain, all deterministic transitions in this model have been replaced by exponential transitions with mean firing delay equal to the constant time.

The GSPN model that we developed simulates the operation of a frame transmitter process driven by SIMNET simulators. Sixteen chains are used to capture the state of transmission of sixteen attempts for each node. A packet is discarded if it is not successfully transmitted in sixteen attempts. In addition, the backoff delay after an unsuccessful transmission is drawn from different distributions for different attempts as specified by the IEEE 802.3 collision resolution protocol. In this model, we accurately represent the frame transmitter process keeping track of all sixteen attempts separately and using the correct backoff delay distribution for each attempt. In the following, we use *i* ,where *i* varies from 0 to 15 to designate actions and states for the *(i+1)*th attempt.

The data transmission can be broken down into five phases: (1) Initial transmission phase, (2) Transmit listen phase, (3) Successful transmission phase, (4) Collision abort and prepare for next attempt phase, and (5) Reinitialization after collision phase.

Initial Transmission Phase: This initial transmission phase is given in Figure 1. The channel is assumed to be free initially. Any node ready to transmit can transmit at this time. The node is indicated by a token in place *Potential(i)* where *i* indicates that this is the *(i+1)*th attempt by the node. The start of the transmission is indicated by firing the timed transition *Transmit-Period(i)*. The firing of this transition deposits one token in *Transmitting(i)* and another token in *Count*. A token in *Transmitting* indicates that a node is transmitting a frame. This enables and fires the immediate transition *First-Transmit(i)* if there is no token in *First-Node*. A token in *First-Node* indicates the first node attempting to transmit a message. The number of tokens in *Count* indicates the number of nodes transmitting frames simultaneously. Thus more than one token in *Count* would indicate that a collision is going to occur during this transmission, which is modeled by firing the immediate transition *Collision-Detected*. The transition *Collision-Detected* puts a token in *Collide* indicating the condition of a collision. In the final model, the immediate transition *Collision-Detected* was replaced by a timed transition with a very high firing rate to determine the expected number of collisions.

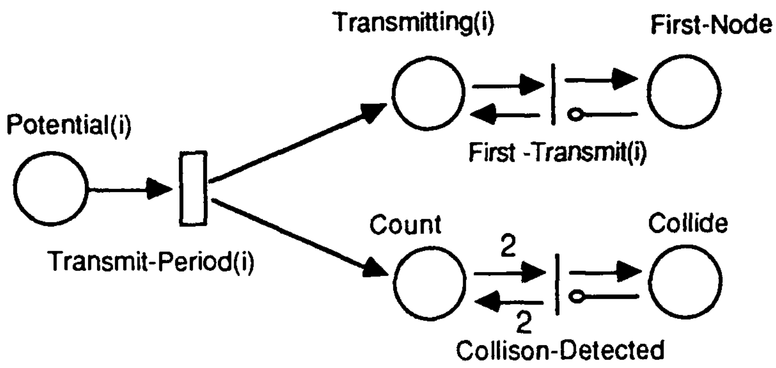


Figure 1. Initial Transmission Phase [(i+1)th attempt]

Transmit Listen Phase: This phase starts as soon as tokens are deposited in *Transmitting(i)* and *First-Node* in the initial transmission phase. In this phase, a node is listening while transmitting a frame to detect whether there is a collision. If there is a collision, it is detected within a time period dependent on the distance between the two nodes transmitting simultaneously and the number of segments in the Ethernet. If no collision is detected within this time period, then it is assumed that the node has acquired the channel and the transmission of the frame will be completed without collision. This is shown in Figure 2. The timed transition *Listen-Period* indicates the time period for collision detection. Thus a token in *Detect* would indicate that it is time to detect the collision or no collision situation . This is modeled by firing one of the immediate transitions *No-Collision* and *Collision.* A token in *Acquire* would indicate that the channel has been acquired and no collision will occur. A token in *Jam* indicates collision has been detected and jam signal should be sent to abort the frame transmission.

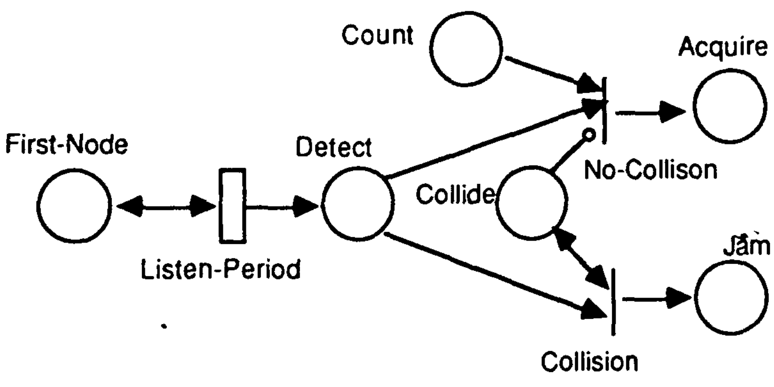


Figure 2. Transmit Listen Phase

Successful Transmission Phase: Two cases occur in the Transmit Listen Phase: a node is successfully transmitting a frame or there is a collision. In this phase we model the first case. A token in *Acquire* would indicate that the node is successfully transmitting and the remaining time to transmit is modeled by the timed transition *Transmit-time*. When this transition fires it is assumed that the data transfer is complete and a token is deposited in *Trans-Done* which in turn enables the immediate transition *Complete(i)*. This latter transition fires and puts a token in *Potential(0)* to indicate that the node is again ready to transmit a frame if there is one. Figure 3 shows this phase.

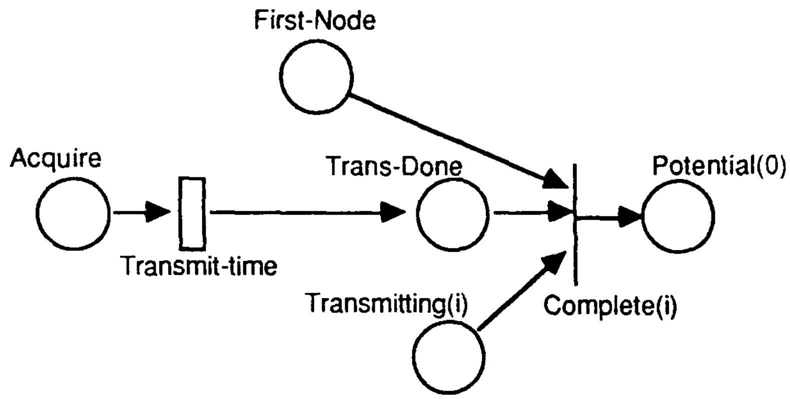


Figure 3. Successful Transmission Phase [(i+1)th attempt]

Collision Abort and Prepare for Next Attempt Phase: In this phase, we model collision detection during the Transmit Listen Phase. Once a collision is detected, a token is put in *Jam* to indicate that a jam signal is transmitted on the channel. The time required by this jam signal is modeled by the timed transition *Jam-Delay*. When this transition fires, a token is deposited in *Abort* indicating that it is time for the channel to be clear so a frame can be transmitted again. The token in *Abort* enables the immediate transition Ready((i+1) mod 16) when at least one node was attempting to send a frame the *(i+1)*th time. The firing of these transitions brings the token count to zero in *Count* and makes a node with *(i+1)*th attempt to be ready for the *(i+2)*th attempt, where *i* is less than 15, by removing a token from Transmitting(i) and depositing a token in Potential((i+1) mod 16). For i=15, a token is removed from Transmitting(15) and a token is deposited in Potential(0) indicating that the frame is lost because of too many collisions and the node is ready to transmit the next frame. This is shown in Figure 4. At this point all nodes are prepared for frame transmission, but several tokens are still present indicating collision. These are removed in the next phase.

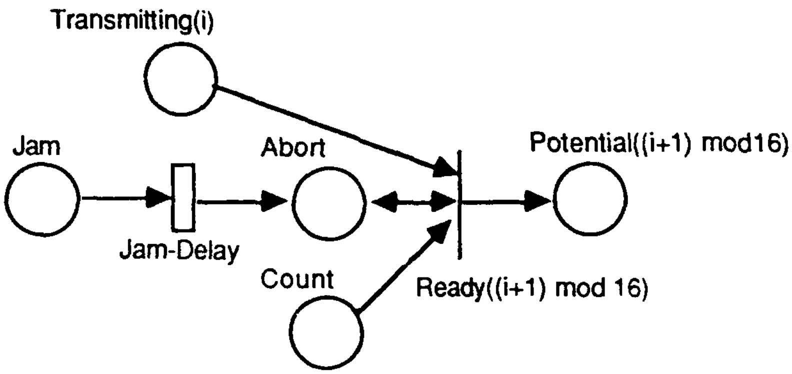


Figure 4. Collision-Abort and Prepare for Next Attempt

Reinitialization After Collision: In this phase, the tokens at *Collide*, *Abort*, and *First-Node* are removed by enabling and firing the immediate transition *Initialization.* This transition is enabled when there is no token in *Count* and there is a token in *Restart* in addition to tokens in *Collide*, *Abort*, and *First-Node*. Thus as soon as all nodes involved in a collision get ready for their next attempt, this transition is fired if *Restart* has a token. Initially, we put a token in *Restart* and it always stays there. Thus the purpose of the place Restart with a token is to reinitialize the model for next transmission. This phase is shown in Figure 5.

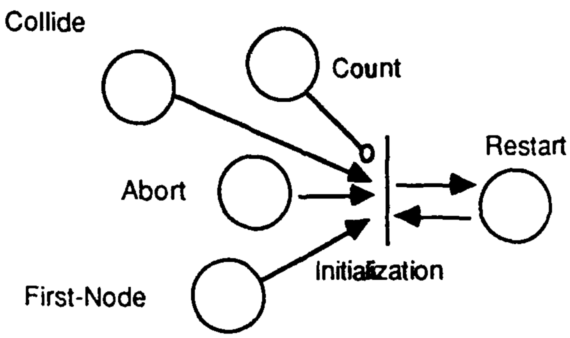


Figure 5. Reinitialization After Collision

Finally, the deference process is modeled via inhibitor arcs from *Acquire* and *Jam* to *Transmit-Period(i)* as well as from *Acquire* and *Jam* to *Listen-Period. Transmit-Period(i)* and *Listen-Period* are respectively given in figure 1 and 2. But these inhibitors arcs are not drawn here although they exist in the model.

We have described global model 1 in phases. Since the diagram of this model is very large we are unable to show it's Petri net diagram. However, at the initial developmental stage, we used GreatSPN [Ch'87] to check the structural properties of the Petri net model when only three attempts are permitted for transmitting a frame. We then used the SPNP package [Ci'88] to develop and execute global model 1. This model, being large, suffers from the practical difficulty in providing analytic solution for a large number of nodes. Specifically, in this case, the number of tangible markings becomes very large and cannot be solved on Sun 3/60 workstation. Global model 1 has been very useful, however, in obtaining results for relatively small networks (with up to 3 stations). It has also been useful in evaluating the accuracy of the approximations employed in the other two models. These approximations are explained next.

Global Model 2:

Global model 2 has three chains compared to sixteen in global model 1. Chain 1 models the operation of nodes in their 1st transmission attempt. Chain 2 indicates operation of nodes in their 2nd through 10th retransmission attempts. Chain 3 stands for the operation of nodes in their 11th through 16th retransmission attempts. The exponential transition in the 2nd chain in global model 2 actually models the weighted average of backoff delays in chains for retransmission attempts 2 to 10 in global model 1. Also the mean delay of the exponential transition in chain 3 is the backoff delays in chains for retransmission attempts 11 to 16 in the previous model. Four random switches with appropriate probabilities are used and are divided into two groups. The two random switches in the first group perform two distinct operations. One of them keeps nodes in chain 2 while the other moves them to the starting place of chain 3. Similarly, one of the two random switches in the 2nd group keeps nodes in chain 3 and the other one transfers them to the starting place of chain 1. With this modified model based on suitable assumption for backoff delays and probabilities of random switches, a substantial reduction in the size of the reachability graph is obtained compared to Global model 1.

Global Model 3:

Global model 3 is a further simplification of global model 2. It has two chains compared to sixteen in global model 1 and three in global model 2. Chain 1 implies operation of stations as in chain 1 of Global model 1 and 2. But chain 2 indicates operation of stations in their second through 16-th retransmission attempts. The exponential transition at the beginning of chain 2 models the weighted average of backoff delays in chains for retransmission attempts 2 to 16 in Global model

1. Two random switches with appropriate probabilities either keep stations in chain 2 or move them to the starting place of chain 1.This model gives further reduction in the size of the reachability graph over Global model 2.

MODEL SIZE AND EXECUTION TIME COMPARISON

All three global models described above were analysed with tools under GreatSPN and SPNP packages.Table 1 gives the comparison of the reachability graph for these three models. Table 2 provides a comparison of the execution time of global model 3 with two different sizes of primary memory.

The binary exponential backoff delay as specified in IEEE 802.3 is approximated by exponential transitions with a mean firing delay equal to the average of binary backoff delays. As we develop modified global models we potentially gain the advantage of analyzing networks with a higher number of stations at the cost of adding a bit more approximation which was observed to be reasonable. All such global models when run under GreatSPN or SPNP give performance parameters which are very close to each other.

An additional observation from solving the global models indicates that with a five segment ETHERNET cable containing repeater electronics, the offered and transmitted load could not exceed a slowly varying threshold. A decrease in the number of cable segments reduced collision and incresed both offered and transmitted load.

No. of Stations	Global Model 1	Global Model 2	Global Model 3
2	T=1192 V=1704 A=3648	T=48 V=66 A=150	T=23 V=31 A=68
3	T= 17852 V= 22576 A=62304	T=168 V=216 A=583	T=60 V=78 A=196
4	T= ?? V= ?? A=??	T=444 V=560 A=1660	T=122 V=158 A=425
5	T= ?? V= ?? A=??	T=981 V=1233 A=3903	T=214 V=278 A=779
6	T=?? V=?? A=??	T=1919 V=2417 A=7993	T=341 V=445 A=1282
7	T=?? V=?? A=??	T=3438 V=4348 A=14862	T=508 V=666 A=1958
8	T=?? V=?? A=??	T=5763 V=7323 A=25695	T=720 V=948 A=2831
9	T= ?? V= ?? A=??	T=9169 V=11707 A=41968	T=982 V=1298 A=3925
10	T= ?? V= ?? A=??	T=13986 V=17940 A=65480	T=1299 V=1723 A=5264

Table 1. Comparison of the reachability graph of the three global models
T= tangible marking, V= vanishing marking, ??= not attempted
A= Number of Arcs

No. of Nodes	Tangible Marking	Vanish Markings	Arcs	Execution 4 MB RAM	Execution 16 MB RAM
20	3578	3601	13108	14 mins	15mins
25	6598	6626	24573	46 mins	41 mins
30	10968	11001	41313	135 mins	107 mins
35	16938	16976	64328	278 mins	253 mins
40	24578	24801	94618	2062 mins	573 mins

Table 2. Reachability graph and execution time with two different size of RAMs

PERFORMANCE INDICES

The basis for computation of performance indices of the GSPN model of the SIMNET protocol is the steady state probabilities of the markings. We derive these marking probabilities using the SPNP package. The expected number of tokens at any place is given by E(parameter) where parameter=mark(<place-name>) and E is the expectation operator. Similarly, the expected firing rate of an exponential transition is given by E(parameter) where parameter=rate(<transition-name>). This is the same as the definition <firing-rate>* Pr{condition that transition is enabled}.

We now introduce the performance indices for the GSPN model. Since the global model does not incorporate the details of each station separately, the performance parameters can only be defined for the entire net. Although each successive model is more compact than the previous one, each loses some information about the reattempts. For example, in model 1 we could compute the average number of stations waiting for all sixteen different attempt groups.But in model 2 we can compute only the number of stations in the three groups: 1st attempt, 2nd through 10th attempt and 11th through 16th attempt. Similarly in model 3 we can compute the number of stations in two groups only, namely 1st attempt and 2nd through 16th attempt. We define the important network parameters as Load, Throughput, Dataloss, Utilization as below:

Load= E(rate(*TransmitPeriod(i)*)), the average number of input messages to the network per unit time;

Throughput=E(rate(*Transmit-time*)), the average number of successful message transmission per unit time;

Dataloss= Throughput - Load, the number of message loss per unit time due to collision;

Utilization= parameter1 *E(rate(*Collision-Detected*)) + parameter2 * expected(rate(*Transmit-time*)), where parameter1= Jam Delay + Partially Transmitted Frame Delay + Interframe Spacing Delay, parameter2=Frame Transmission delay + Interframe Spacing Delay, and the utilization is the bandwidth used for both unsuccessful and successful transmission.

In the next section numerical results obtained from the model using the above definitions are provided.

PERFORMANCE RESULTS

Results pertaining to the performance indices of the Petri net models are presented. Within the background of SIMNET we have evaluated network performance under Ethernet protocols. Clusters of 10, 20, and 30 nodes have been considered. Each node is a multi-port transceiver used to connect eight SIMNET nodes to a single point on the coaxial cable.The analysis has been carried out for one through five segments of Ethernet with frame size of 2048 bits, repeater electronics delays, jam signal and other overhead traffic delays. Each station uses an exponential stochastic distribution for packet arrivals with a cumulative rate that drives the network at various levels of total traffic load. In the Petri net models, all deterministic time slices have been approximated by exponential distributions having a mean equal to the deterministic time slice. The speed of data propagation in the coaxial cable is assumed to be the speed of light.

The results presented here are based on global model 3. Throughput, utilization and data loss are shown in Table 3. Table 4 gives the expected number of tokens at various places of the Petri net model.

The graphical plots have been done for the relationships (1) Throughput vs analytical load, (2). Utilization vs analytical load, and (3) Data Loss vs analytical load; and are given in the Appendix. The analytical load is defined as the load (number of packets) obtained if the entire number of nodes (simulators) can always (i.e. all the time) generate new packets. Graphs 1 through 15 compare these results for 1 through 5 segments for a given number of clusters of stations. Graphs 16 through 30 compare the results for 10, 20, 30, 35 and 40 clusters of stations for a given number of segments.

Cluster of Nodes = 10, 20, 30, 35, 40. No. of Segments = 01. Channel Capacity = 10 Mbits/Sec.
Slot Time = 51.2 MicroSeconds. Packet Size = 2048 bits.

No. of Clus-ters	Mess./ Sec/ Stn	Offered Load (Mbit/Sec)	Throughput (Mbit/Sec)	Data Loss. (Pac/Sec)	Utilization (Mbit/Sec)
10	10	0.2007	0.2007	0.0000	0.2101
10	50	0.9293	0.9293	0.0000	0.9729
10	100	1.7009	1.7009	0.0008	1.7807
10	500	5.0297	5.0297	0.0334	5.2678
10	1000	6.5551	6.5548	0.1563	6.8680
20	10	0.3935	0.3935	0.0000	0.4120
20	50	1.7014	1.7014	0.0008	1.7813
20	100	2.9088	2.9088	0.0025	3.0457
20	500	6.6269	6.6266	0.1863	6.9437
30	10	0.5790	0.5790	0.0000	0.6062
30	50	2.3531	2.3531	0.0020	2.4638
30	100	3.8107	3.8107	0.0108	3.9905
30	500	7.2794	7.2786	0.4018	7.6299
35	10	0.6691	0.6691	0.0000	0.7005
35	50	2.6422	2.6422	0.0026	2.7666
35	100	4.1810	4.1810	0.0144	4.3785
35	500	6.8763	7.0085	0.4690	7.3456
35	1000	7.3343	7.3566	0.4940	7.7122
35	5000	7.4997	7.4974	0.4990	7.8598
40	10	0.7575	0.7575	0.0001	0.7931
40	50	2.9104	2.9104	0.0036	3.0475
40	100	4.5096	4.5096	0.0242	4.7229
40	500	7.4126	7.4116	0.4890	7.7703
40	1000	7.4241	7.4231	0.4951	7.7823
40	5000	7.4959	7.4948	0.4990	7.8571

Table 3a. Results for 10,20,30, 35 and 40 Clusters with 1 segment

Cluster of Nodes = 10, 20,30 , 35, 40. No. of Segments = 02. Channel Capacity = 10 Mbits/Sec.
Slot Time = 51.2 MicroSeconds. Packet Size = 2048 bits.

No. of Clus-ters	Mess./ Sec/ Stn	Offered Load (Mbit/Sec)	Throughput (Mbit/Sec)	Data Loss. (Pac/Sec)	Utilization (Mbit/Sec)
10	10	0.2009	0.2009	0.0001	0.2103
10	50	0.9328	0.9328	0.0078	0.9771
10	100	1.7083	1.7082	0.0459	1.7904
10	500	4.5948	4.5910	1.8144	4.8315
10	1000	4.9484	4.9425	2.8884	5.2068
10	5000	5.2756	5.2694	3.0502	5.5477
20	10	0.3944	0.3944	0.0009	0.4130
20	50	1.7136	1.7134	0.0525	1.7960
20	100	2.9260	2.9253	0.3134	3.0703
20	500	4.9182	4.9122	2.9233	5.1755
20	1000	4.9529	4.9469	2.9340	5.2117
20	5000	5.2756	5.2694	3.0501	5.5477
30	10	0.5809	0.5809	0.0026	0.6083
30	50	2.3758	2.3754	0.1576	2.4917
30	100	3.8199	3.8180	0.9081	4.0128
30	1000	4.9529	4.9469	2.9355	5.2110
30	5000	5.2756	5.2694	3.0496	5.5477
35	10	0.6716	0.6716	0.0037	0.7034
35	50	2.6702	2.6697	0.2392	2.8013
35	100	4.1665	4.1665	0.3710	4.3821
35	500	4.9184	4.9123	2.9339	5.1756
35	1000	4.9529	4.9469	2.9309	5.2117
35	5000	5.2756	5.2694	3.0492	5.5477
40	10	0.7608	0.7607	0.0051	0.7968
40	50	2.9433	2.9426	0.3397	3.0887
40	100	4.4559	4.4522	1.8003	4.6855
40	500	4.9183	4.9123	2.9243	5.1756
40	1000	4.9529	4.9469	2.9358	5.2117
40	5000	5.2756	5.2694	3.0493	5.5477

Table 3b. Results for 10,20,30 ,35 and 40 Clusters with 2 segments

Cluster of Nodes = 10, 20,30 , 35, 40. No. of Segments = 03. Channel Capacity = 10 Mbits/Sec.
Slot Time = 51.2 MicroSeconds. Packet Size = 2048 bits.

No. of Clus-ters	Mess./ Sec/ Stn	Offered Load (Mbit/Sec)	Throughput (Mbit/Sec)	Data Loss. (Pac/Sec)	Utilization (Mbit/Sec)
10	10	0.2011	0.2011	0.0006	0.2106
10	50	0.9363	0.9362	0.0287	0.9816
10	100	1.7146	1.7142	0.1660	1.8000
10	500	4.0675	4.0591	4.0940	4.2968
10	1000	4.1740	4.1643	4.6968	4.4101
10	5000	4.6436	4.6333	5.0060	4.8986
20	10	0.3952	0.3952	0.0033	0.4140
20	50	1.7250	1.7246	0.1903	1.8113
20	100	2.9278	2.9255	1.1205	3.0827
20	500	4.1272	4.1176	4.6756	4.3614
20	1000	4.1741	4.1645	4.7043	4.4103
20	5000	4.6436	4.6333	5.0057	4.8986
30	10	0.5828	0.5827	0.0093	0.6106
30	50	2.3954	2.3942	0.5768	2.5190
30	100	3.7349	3.7288	2.9991	3.9421
30	1000	4.1741	4.1645	4.7030	4.4103
30	5000	4.6436	4.6333	5.0052	4.8986
35	10	0.6741	0.6741	0.0128	0.7065
35	50	2.6923	2.6905	0.8745	2.8334
35	100	3.9373	3.9430	2.7903	4.1734
35	500	4.1272	4.1176	4.6709	4.3614
35	1000	4.1741	4.1645	4.7029	4.4103
35	5000	4.6436	4.6333	5.0055	4.8986
40	10	0.7640	0.7639	0.0184	0.8008
40	50	2.9657	2.9631	1.2483	3.1233
40	100	4.0634	4.0542	4.4665	4.2940
40	500	4.1272	4.1176	4.6749	4.3614
40	1000	4.1741	4.1645	4.7028	4.4103
40	5000	4.6436	4.6333	5.0053	4.8986

Table 3c. Results for 10,20,30 ,35 and 40 Clusters with 3 segments

Cluster of Nodes = 10, 20, 30, 35, 40. No. of Segments = 04. Channel Capacity = 10 Mbits/Sec.
Slot Time = 51.2. MicroSeconds. Packet Size = 2048 bits.

No. of Clus-ters	Mess./ Sec/ Stn	Offered Load (Mbit/Sec)	Throughput (Mbit/Sec)	Data Loss. (Pac/Sec)	Utilization (Mbit/Sec)
10	10	0.2013	0.2013	0.0013	0.2108
10	50	0.9397	0.9395	0.0626	0.9863
10	100	1.7196	1.7189	0.3635	1.8095
10	500	3.6494	3.6373	5.8856	3.8750
10	1000	3.7173	3.7047	6.1884	3.9466
10	5000	4.2901	4.2763	6.7244	4.5419
20	10	0.3961	0.3961	0.0074	0.4151
20	50	1.7357	1.7348	0.4226	1.8271
20	100	2.9088	2.9039	2.3855	3.0771
20	500	3.6623	3.6497	6.1442	3.8894
20	1000	3.7173	3.7047	6.1887	3.9466
20	5000	4.2901	4.2763	6.7248	4.5419
30	10	0.5846	0.5846	0.0204	0.6131
30	50	2.4105	2.4079	1.2780	2.5445
30	100	3.5306	3.5196	5.3419	3.7479
30	500	3.6623	3.6497	6.1431	3.8894
30	1000	3.7173	3.7047	6.1888	3.9466
30	5000	4.2901	4.2763	6.7247	4.5419
35	10	0.6767	0.6766	0.0303	0.7098
35	50	2.7050	2.7010	1.9206	2.8593
35	100	3.6066	3.5949	5.6710	3.8312
35	500	3.6623	3.6497	6.1426	3.8894
35	1000	3.7173	3.7047	6.1886	3.9466
35	5000	4.2901	4.2763	6.7246	4.5419
40	10	0.7672	0.7671	0.0428	0.8050
40	50	2.9691	2.9693	2.7254	3.1426
40	100	3.6196	3.6071	6.1005	3.8448
40	500	3.6623	3.6497	6.1423	3.8894
40	1000	3.7173	3.7047	6.1886	3.9466
40	5000	4.2901	4.2763	6.7250	4.5419

Table 3d. Results for 10,20 ,30 ,35 and 40 Clusters with 4 segments

Cluster of Nodes = 10, 20, 30, 35, 40. No. of Segments = 05. Channel Capacity = 10 Mbits/Sec.
Slot Time = 51.2. MicroSeconds. Packet Size = 2048 bits.

No. of Clus-ters	Mess./ Sec/ Stn	Offered Load (Mbit/Sec)	Throughput (Mbit/Sec)	Data Loss. (Pac/Sec)	Utilization (Mbit/Sec)
10	10	0.2015	0.2015	0.0024	0.2111
10	50	0.9430	0.9428	0.1110	0.9913
10	100	1.7232	1.7219	0.6381	1.8189
10	500	3.3361	3.3211	7.3223	3.5628
10	1000	3.4004	3.3851	7.5038	3.6301
10	5000	4.0476	4.0307	8.2289	4.3024
20	10	0.3969	0.3969	0.0131	0.4163
20	50	1.7454	1.7438	0.7524	1.8433
20	100	2.8652	2.8570	3.9853	3.0488
20	500	3.3394	3.3242	7.4401	3.5666
20	1000	3.4004	3.3851	7.5063	3.6301
20	5000	4.0476	4.0307	8.2882	4.3024
30	10	0.5865	0.5864	0.0360	0.6157
30	50	2.4192	2.4145	2.2677	2.5664
30	100	3.2763	3.2614	7.2703	3.4993
30	500	3.3394	3.3242	7.4403	3.5666
30	1000	3.4004	3.3851	7.5059	3.6301
30	5000	4.0476	4.0307	8.2889	4.3024
35	10	0.6792	0.6791	0.0533	0.7133
35	50	2.7032	2.6965	3.2748	2.8740
35	100	3.2924	3.2773	7.4079	3.5174
35	500	3.3394	3.3242	7.4404	3.5666
35	1000	3.4004	3.3851	7.5058	3.6301
35	5000	4.0476	4.0307	8.2889	4.3024\
40	10	0.7705	0.7703	0.0754	0.8095
40	50	2.9424	2.9329	4.6423	3.1333
40	100	3.2932	3.2780	7.3906	3.5184
40	500	3.3394	3.3242	7.4405	3.5666
40	1000	3.4004	3.3851	7.5055	3.6301
40	5000	4.0476	4.0307	8.2889	4.3024

Table 3e. Results for 10,20 ,30,35 and 40 Clusters with 5 segments

Expected Number of Tokens at Places Potential(0), Potential(1), Acquire, Count, Collide
Cluster of Nodes = 10,20,30,35,40. No. of Segments = 01. P(0)= Potential(0).
P(1)= Potential(1).
Rate implies Message/Sec/Node

Clusters	Rate	E(P(0))	E(P(1))	E(Acquire)	E(Count)	E(Collide))
10	10	9.97985	0.00007	0.01991	0.00015	0.00000
10	50	9.90530	0.00175	0.09220	0.00074	0.00000
10	100	9.82294	0.00692	0.16876	0.00136	0.00001
10	500	9.32655	0.16996	0.49904	0.00443	0.00025
10	1000	8.62224	0.72100	0.65036	0.00639	0.00063
10	5000	2.67850	6.56967	0.74363	0.00819	0.00115
20	10	19.9603	0.00030	0.03905	0.00030	0.00000
20	50	19.8223	0.00743	0.16881	0.00136	0.00001
20	100	19.6792	0.02973	0.28861	0.00239	0.00006
20	500	18.3486	0.98725	0.65748	0.00657	0.00069
20	1000	13.4172	5.84029	0.73429	0.00812	0.00116
20	5000	2.67860	16.56950	0.74363	0.00819	0.00115
30	10	29.9414	0.00068	0.05745	0.00045	0.00000
30	50	29.7474	0.01713	0.23347	0.00191	0.00003
30	100	29.5487	0.06992	0.37809	0.00322	0.00012
30	500	25.2561	4.01381	0.72218	0.00785	0.00107
30	1000	13.6342	15.62100	0.73651	0.00819	0.00116
30	5000	2.67860	26.56950	0.74363	0.00819	0.00115
35	10	34.8980	0.03324	0.05093	0.01777	0.00039
35	50	32.9875	1.72489	0.20224	0.08527	0.00851
35	100	21.3466	13.2947	3.27730	7.40791	3.51748
35	500	4.30942	30.3273	0.24931	0.11387	0.01442
35	1000	2.17983	32.4510	0.25388	0.11521	0.01439
35	5000	0.48686	34.0813	0.30230	0.12944	0.01379
40	10	39.9230	0.00123	0.07516	0.00059	0.00000
40	50	39.6778	0.03101	0.28877	0.00240	0.00006
40	100	39.4184	0.13023	0.44744	0.00392	0.00020
40	500	27.2731	11.98327	0.73537	0.00819	0.00118
40	1000	13.6343	25.62091	0.73651	0.00819	0.00118
40	5000	2.6786	36.56956	0.74363	0.00819	0.00115

Table 4a. Expected Number of Tokens at various Places for 1 segment

Expected Number of Tokens at Places Potential(0), Potential(1), Acquire, Count, Collide
Cluster of Nodes = 10,20,30,35,40. No. of Segments = 02. P(0)= Potential(0).
P(1)= Potential(1).
Rate implies Message/Sec/Node

Clusters	Rate	E(P(0))	E(P(1))	E(Acquire)	E(Count)	E(Collide))
10	10	9.97927	0.00062	0.01871	0.00138	0.00000
10	50	9.89078	0.01577	0.08690	9.00653	0.00007
10	100	9.76444	0.06414	0.15914	0.01225	0.00028
10	500	7.61717	1.91669	0.42772	0.03840	0.00342
10	1000	4.38482	5.11179	0.46046	0.04291	0.00441
10	5000	0.88535	8.57901	0.49092	0.04470	0.00423
20	10	19.9578	0.00266	0.03674	0.00272	0.00001
20	50	19.7579	0.07008	0.15963	0.01233	0.00030
20	100	19.3892	0.31608	0.27254	0.02215	0.00106
20	500	8.78665	10.7128	0.45764	0.04282	0.00446
20	1000	4.39738	15.0987	0.46088	0.04300	0.00444
20	5000	0.88535	18.5790	0.49092	0.04470	0.00423
30	10	29.9357	0.00613	0.05412	0.00403	0.00003
30	50	29.5879	0.17313	0.22131	0.01756	0.00065
30	500	8.78871	20.7124	0.45765	0.04282	0.00446
30	1000	4.39740	25.0987	0.46088	0.04300	0.00444
30	5000	0.88535	28.5790	0.49092	0.04470	0.00423
35	10	34.9243	0.00842	0.06257	0.00467	0.00004
35	50	34.4826	0.24861	0.24872	0.02001	0.00087
35	100	32.9780	1.59966	0.38817	0.03409	0.00273
35	500	8.78720	25.7122	0.45765	0.04282	0.00446
35	1000	4.39736	30.0987	0.46087	0.04300	0.00444
35	5000	0.88535	33.5790	0.49092	0.04470	0.00423
40	10	39.9127	0.01108	0.07087	0.00531	0.00005
40	50	39.3585	0.34492	0.27414	0.02236	0.00110
40	100	36.9508	2.59710	0.41479	0.03727	0.00332
40	500	8.78712	30.7123	0.45765	0.04282	0.00446
40	1000	4.39740	35.0987	0.46088	0.04300	0.00448
40	5000	0.88535	38.5790	0.49092	0.04470	0.00423

Table 4b. Expected Number of Tokens at various Places for 2 segments

Expected Number of Tokens at Places Potential(0), Potential(1), Acquire, Count, Collide
Cluster of Nodes = 10,20,30,35,40. No. of Segments = 03. P(0)= Potential(0).
P(1)= Potential(1).
Rate implies Message/Sec/Node

Clusters	Rate	E(P(0))	E(P(1))	E(Acquire)	E(Count)	E(Collide))
10	10	9.97867	0.00119	0.01752	0.00261	0.00000
10	50	9.87534	0.03058	0.08155	0.01251	0.00022
10	100	9.69901	0.12790	0.14933	0.02375	0.00081
10	500	6.05253	3.52662	0.35359	0.06724	0.00715
10	1000	3.12964	6.43799	0.36275	0.06960	0.00762
10	5000	0.65304	8.86868	0.40361	0.07466	0.00724
20	10	19.9553	0.00509	0.03443	0.00517	0.00003
20	50	19.6830	0.14265	0.15023	0.02400	0.00087
20	100	18.9763	0.72487	0.25484	0.04391	0.00302
20	500	6.23092	13.3412	0.35869	0.06911	0.00765
20	1000	3.12995	16.4376	0.36276	0.06961	0.00762
20	5000	0.65304	18.8686	0.40361	0.07466	0.00724
30	10	29.9297	0.01182	0.05076	0.00768	0.00008
30	50	29.3766	0.38006	0.20856	0.03473	0.00189
30	100	26.8519	2.76307	0.32481	0.06016	0.00581
30	500	6.23091	23.3412	0.35869	0.06911	0.00765
30	1000	3.12995	26.4376	0.36276	0.06961	0.00762
30	5000	0.65304	28.8686	0.40361	0.07466	0.00724
35	10	34.9160	0.01629	0.05875	0.00893	0.00012
35	50	34.1513	0.57445	0.23437	0.03984	0.00252
35	100	29.2124	5.37886	0.34348	0.06517	0.00686
35	500	6.23087	28.3413	0.35868	0.06911	0.00765
35	1000	3.12995	31.4376	0.36276	0.06961	0.00762
35	5000	0.65304	33.8686	0.40361	0.07466	0.00724
40	10	39.9017	0.02152	0.06655	0.01016	0.00015
40	50	38.8484	0.84860	0.25812	0.04480	0.00321
40	100	30.6770	8.90179	0.35316	0.06797	0.00749
40	500	6.23092	33.3412	0.35869	0.06911	0.00765
40	1000	3.12995	36.4376	0.36276	0.06961	0.00762
40	5000	0.65304	38.8686	0.40361	0.07466	0.00724

Table 4c. Expected Number of Tokens at various Places for 3 segments

Expected Number of Tokens at Places Potential(0), Potential(1), Acquire, Count, Collide
Cluster of Nodes = 10,20,30,35,40. No. of Segments = 04. P(0)= Potential(0).
P(1)= Potential(1).
Rate implies Message/Sec/Node

Clusters	Rate	E(P(0))	E(P(1))	E(Acquire)	E(Count)	E(Collide))
10	10	9.97806	0.00176	0.01632	0.00385	0.00001
10	50	9.85894	0.04622	0.07615	0.01867	0.00043
10	100	9.62628	0.19853	0.13932	0.03585	0.00161
10	500	5.00164	4.61189	0.29482	0.09163	0.01076
10	1000	2.53874	7.06770	0.30028	0.09326	0.01094
10	5000	0.54871	9.00204	0.34661	0.10261	0.01043
20	10	19.9526	0.00759	0.03210	0.00765	0.00007
20	50	19.5963	0.22657	0.14062	0.03643	0.00175
20	100	18.4133	1.28397	0.23538	0.06734	0.00593
20	500	5.03596	14.5758	0.29583	0.09237	0.01098
20	1000	2.53875	17.0676	0.30028	0.09326	0.01094
20	5000	0.54871	19.0020	0.34661	0.10261	0.01043
30	10	29.9234	0.01775	0.04738	0.01142	0.00017
30	50	29.0968	0.65453	0.19517	0.05347	0.00381
30	100	24.0797	5.54686	0.28528	0.08809	0.01013
30	500	5.03595	24.5758	0.29583	0.09237	0.01098
30	1000	2.53876	27.0676	0.30028	0.09326	0.01094
30	5000	0.54871	29.0020	0.34661	0.10261	0.01043
35	10	34.9073	0.02455	0.05484	0.01329	0.00023
35	50	33.6721	1.04719	0.21893	0.06168	0.00507
35	100	24.8707	9.74673	0.29139	0.09112	0.01086
35	500	5.03595	29.5758	0.29583	0.09237	0.01098
35	1000	2.53876	32.0676	0.30028	0.09326	0.01094
35	5000	0.54871	34.0020	0.34661	0.10261	0.01043
40	10	39.8900	0.03258	0.06218	0.01515	0.00030
40	50	38.0311	1.65906	0.24021	0.06953	0.00642
40	100	25.0108	14.6051	0.29237	0.09165	0.01099
40	500	5.03595	34.5758	0.29582	0.09237	0.01098
40	1000	2.53876	37.0678	0.30028	0.09326	0.01094
40	5000	0.54871	39.0020	0.34661	0.10261	0.01043

Table 4d. Expected Number of Tokens at various Places for 4segments

Expected Number of Tokens at Places Potential(0), Potential(1), Acquire, Count, Collide
Cluster of Nodes = 10,20,30,35,40. No. of Segments = 05. P(0)= Potential(0).
P(1)= Potential(1).
Rate implies Message/Sec/Node

Clusters	Rate	E(P(0))	E(P(1))	E(Acquire)	E(Count)	E(Collide))
10	10	9.97743	0.00235	0.01511	0.00509	0.00003
10	50	9.84153	0.06272	0.07071	0.02503	0.00072
10	100	9.54592	0.27634	0.12914	0.04857	0.00270
10	500	4.30148	5.33587	0.24908	0.11355	0.01432
10	1000	2.17983	7.45107	0.25388	0.11521	0.01439
10	5000	0.48686	9.08138	0.30230	0.12944	0.01379
20	10	19.9498	0.01016	0.02977	0.01017	0.00012
20	50	19.4961	0.32342	0.13079	0.04965	0.00295
20	100	17.6921	2.00186	0.21428	0.09171	0.00964
20	500	4.30942	15.3273	0.24931	0.11387	0.01442
20	1000	2.17983	17.4510	0.25388	0.11521	0.01439
20	5000	0.48681	19.0813	0.30230	0.12944	0.01379
30	10	29.9168	0.02393	0.04398	0.01524	0.00028
30	50	28.7286	1.01642	0.18109	0.07379	0.00644
30	100	21.1916	8.45196	0.24460	0.11182	0.01417
30	500	4.30942	25.3273	0.24931	0.11387	0.01442
30	1000	2.17983	27.4510	0.25388	0.11521	0.01439
30	5000	0.48686	29.0813	0.30230	0.12944	0.01379
35	10	34.9321	0.00093	0.06639	0.00052	0.00000
35	50	34.7121	0.02353	0.26214	0.00216	0.00005
35	100	34.4842	0.09734	0.41484	0.00359	0.00016
35	500	21.8811	12.4162	0.69538	0.00730	0.00090
35	1000	13.1471	21.1148	0.72991	0.00804	0.00113
35	5000	2.68425	31.5636	0.74389	0.00819	0.00115
40	10	39.8776	0.04430	0.05779	0.02030	0.00051
40	50	36.7703	2.91395	0.21997	0.09571	0.01059
40	100	21.3546	18.2866	0.24585	0.11285	0.01444
40	500	4.30942	35.3273	0.24931	0.11387	0.01442
40	1000	2.17983	37.4510	0.25388	0.11521	0.01439
40	5000	0.48682	39.0813	0.30230	0.12944	0.01379

Table 4e. Expected Number of Tokens at various Places for 5 segments

CONCLUSIONS

In this paper, we presented Petri net models for real-time simulation networks using the Ethernet protocol. Our Petri net models use a global approach in which the interactions and concurrent behavior of all stations on the network are modeled. Unlike isolation models, the global modeling approach is more flexible and accurate, allowing the experimentation with simplifications and approximations.

The numerous tests that we have conducted using different Petri net models show that the throughput of SIMNET with the ETHERNET protocol reaches a maximum of approximately 50-70 % of the transmission medium bandwidth. As explained below, the saturation in throughput is primarily due to the excessive collision rate that characterizes the behaviour of ETHERNETS at high loads. Our results show that ETHERNET is an excellent choice for lightly loaded networks. For example, loads of 10% to 40% of ETHERNET bandwidth represent ideal condition for the ETHERNET medium access protocol (very small collision rate and no packet loss due to excessive collision). With further higher loads, some packets are lost due to exceeding the limit of retransmission attempts, and the performance of ETHERNET rapidly collapses causing packet delays to become too large to be acceptable for real-time simulation.

Petri nets have proven to be a very powerful tool capable of modeling and evaluating the sophisticated logic of the ETHERNET collision handling mechanism and the complex interactions/interferences among the various nodes of a simulation network. This is illustrated in the design of the first global model. This model captures the accurate semantics of operation of the ETHERNET protocol and it reflects a global design that combines detailed modeling of the communication interface of each simulator as well as the asynchronous interactions of the different simulators. This global Petri net model contains adequate logic representing the three transmission phases defined by the MAC sublayer of the IEEE standard 802.3 (namely, the transmitter process, the deference process and the bit transmitter process).

Our experiments have also shown that, within the environment of simulation networks, Petri nets are suitable for the application of hierarchical refinement aimed at reducing the execution time requirements of the models while maintaining the accuracy of the obtained results. This is illustrated in the design of the second and the third global models for SIMNET. The key idea behind these designs is the identification of similar events (processes) that can be accurately integrated into a single Petri net module (reducing the size of the reachability graph for the Petri

net model and consequently reducing the size and space complexity for solving it). Our tests have shown that results obtained througn this hierarchical refinement are quite accurate and the loss in precision due to the aggregation process is not significant. Our tests have also shown that the results obtained by our Petri net models closely agree with the known characteristics and performance features of Ethernet reported in the literature.

REFERENCES

[Ch'87] Chiola, G., "GreatSPN Users' Manual", Universita degli Studi di Torino, 1987.

[Ci'88] Ciardo, G., "Manual for the SPNP Package", Duke University, 1988.

[CiMu'89] Ciardo, G., Muppala, J. Trivedi, K., "SPNP: Stochastic Petri Net Package", Duke University, 1989.

[FrHa '88] Friedman, D. and Haimo, V., "SIMNET ETHERNET Performance", BBN Technical Report No. 6711, 1988.

[Go'85] Gonsalves, T. A., "Performance Characteristics of 2 Ethernets: an Experimental Study", Proceedings of ACM SIGMETRICS Conference on Measurements and Modeling of Computer Systems, pp 78-86, 1985.

[Gr '85] E. Gressier, "A stochastic Petri net model for Ethernet", Proceedings of International Workshop on Timed Petri Nets, pp 296- 303, 1985.

[MaBa '86] Marsan, M. A., Balbo, G. and Conte, G., "Performance Models of Multiprocessor Systems", The MIT Press, 1986.

[MaCh '87] Marsan, M. A., Chiola, G. and Fumagalli, A , "An Accurate Performance Model of CSMA/CD bus LAN", Lecture Notes in Computer Science : Advances in Petri Nets 1987, Vol. 266, pp 146-161, Springer Verlag 1987.

[Po '89] Pope, A. "The SIMNET Network and Protocols", BBN Technical Report No. 7102, 1989.

[Ie '85] IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, ANSI/IEEE Std 802.3-1985, IEEE, 1985.

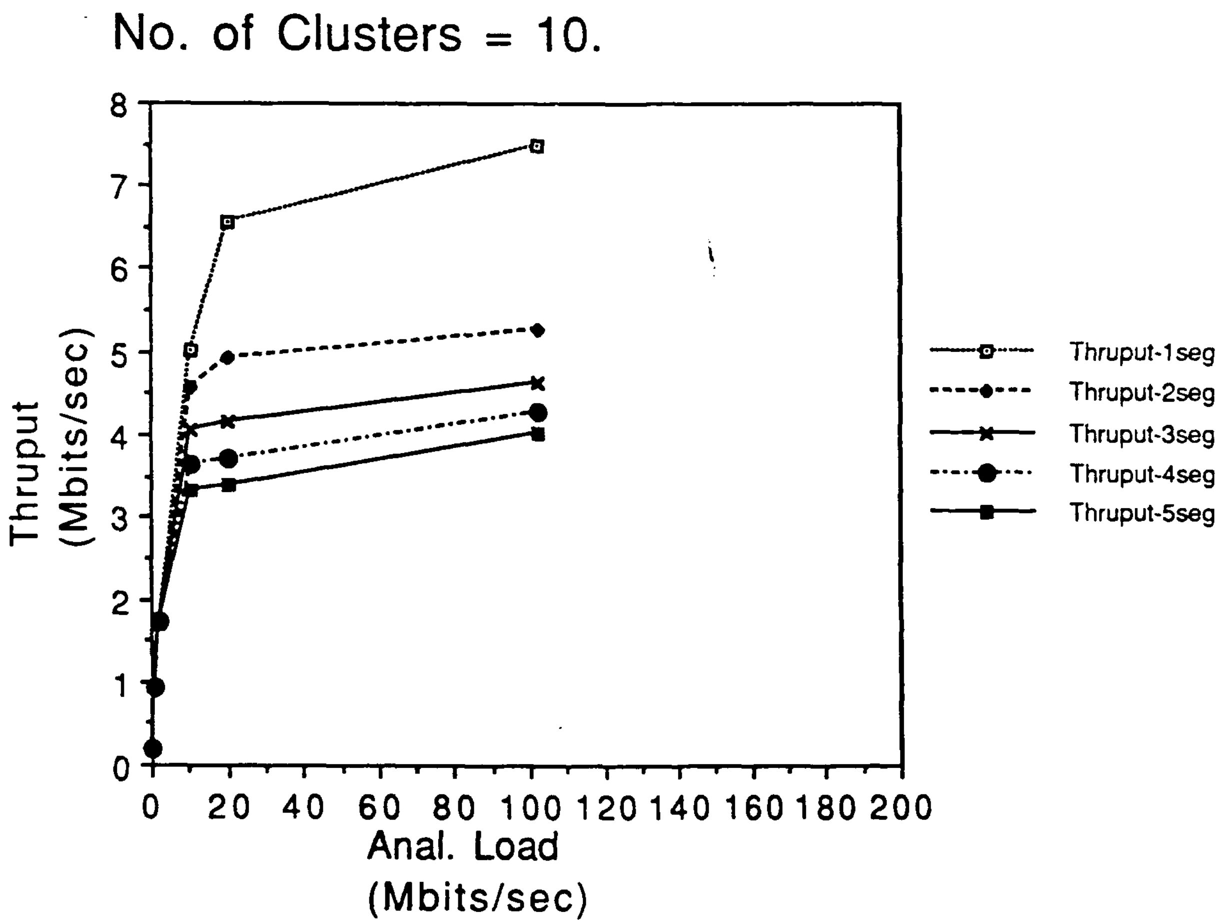
[Ie '87] IEEE Standards for Local Area Networks: Supplements to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, ANSI/IEEE Std 802.3a,b,c,e-1988, IEEE, 1987.

[ToHu '80] Tobagi, F.A. and Hunt, V.B., "Performance Analysis of Carrier Sense Multiple Access with Collision Detection", Computer Networks, Vol 4, pp 245-259 (1980).

[TaKl'85] Takagi, H. and Kleinrock, "Throughput Analysis of Persistent CSMA Systems", IEEE Transactions on Communications, Vol. COM-33, pp 627-638, 1985.

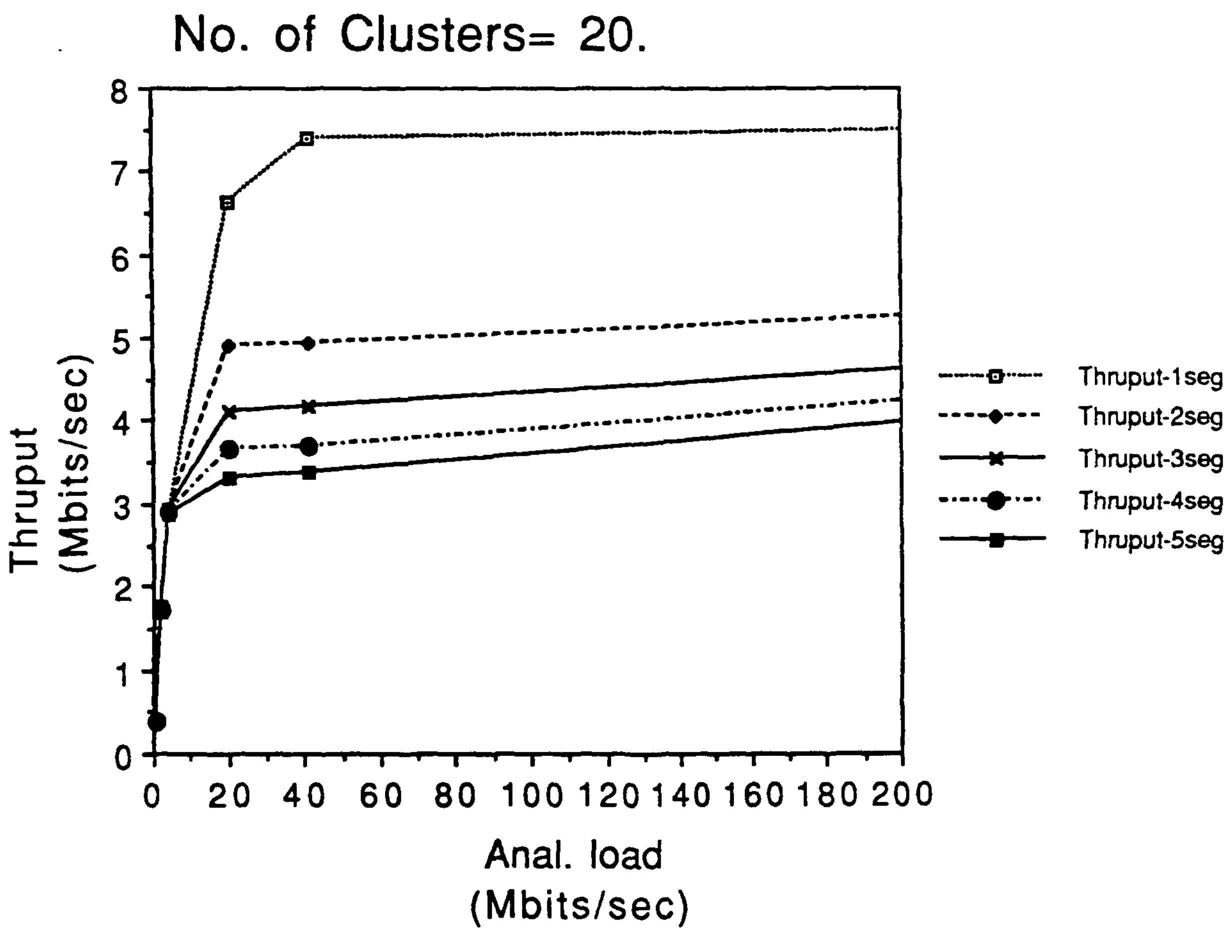
Appendix

Graph 1



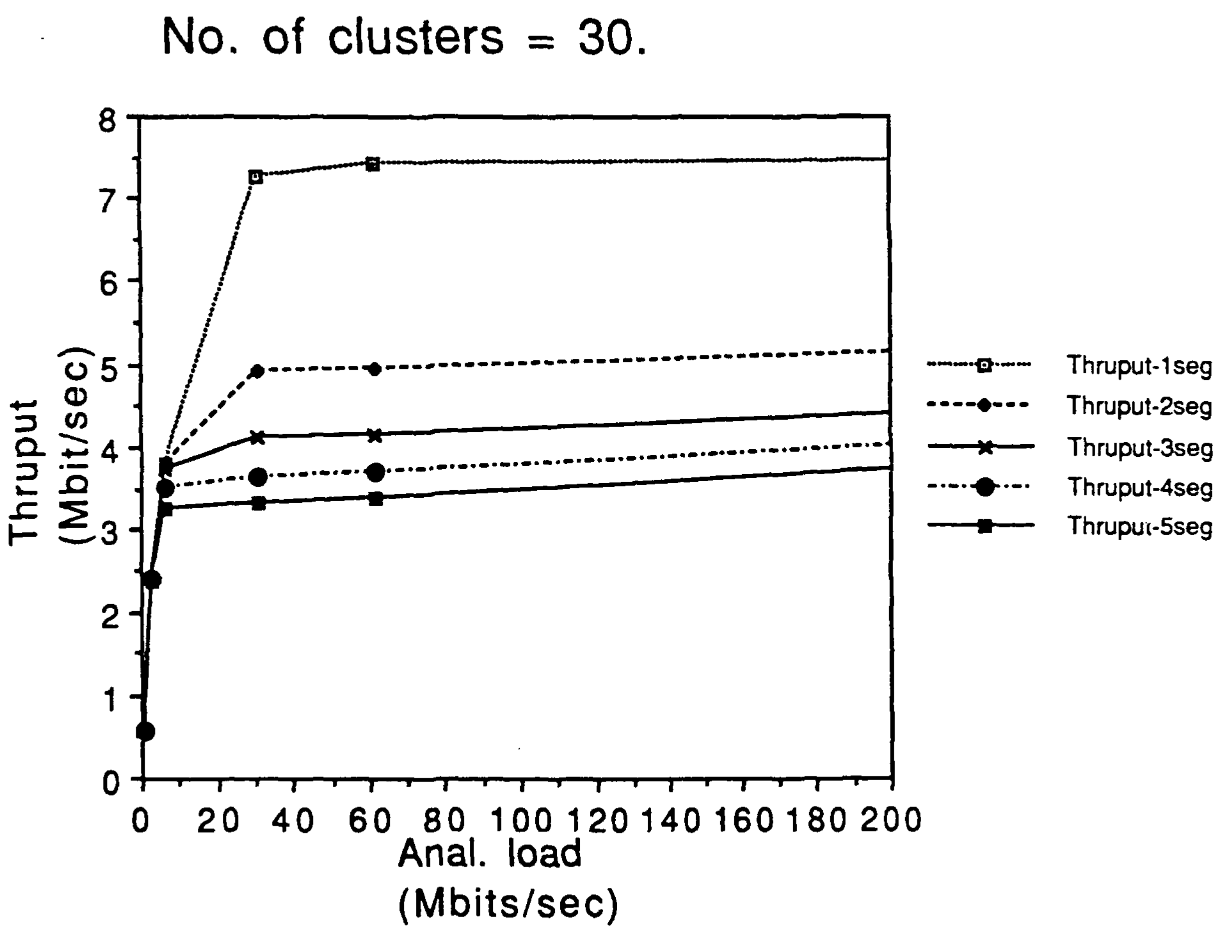
No. of Clusters = 10.
Thruput
(Mbits/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. Load
(Mbits/sec)
Thruput-1seg
Thruput-2seg
Thruput-3seg
Thruput-4seg
Thruput-5seg

Graph 2

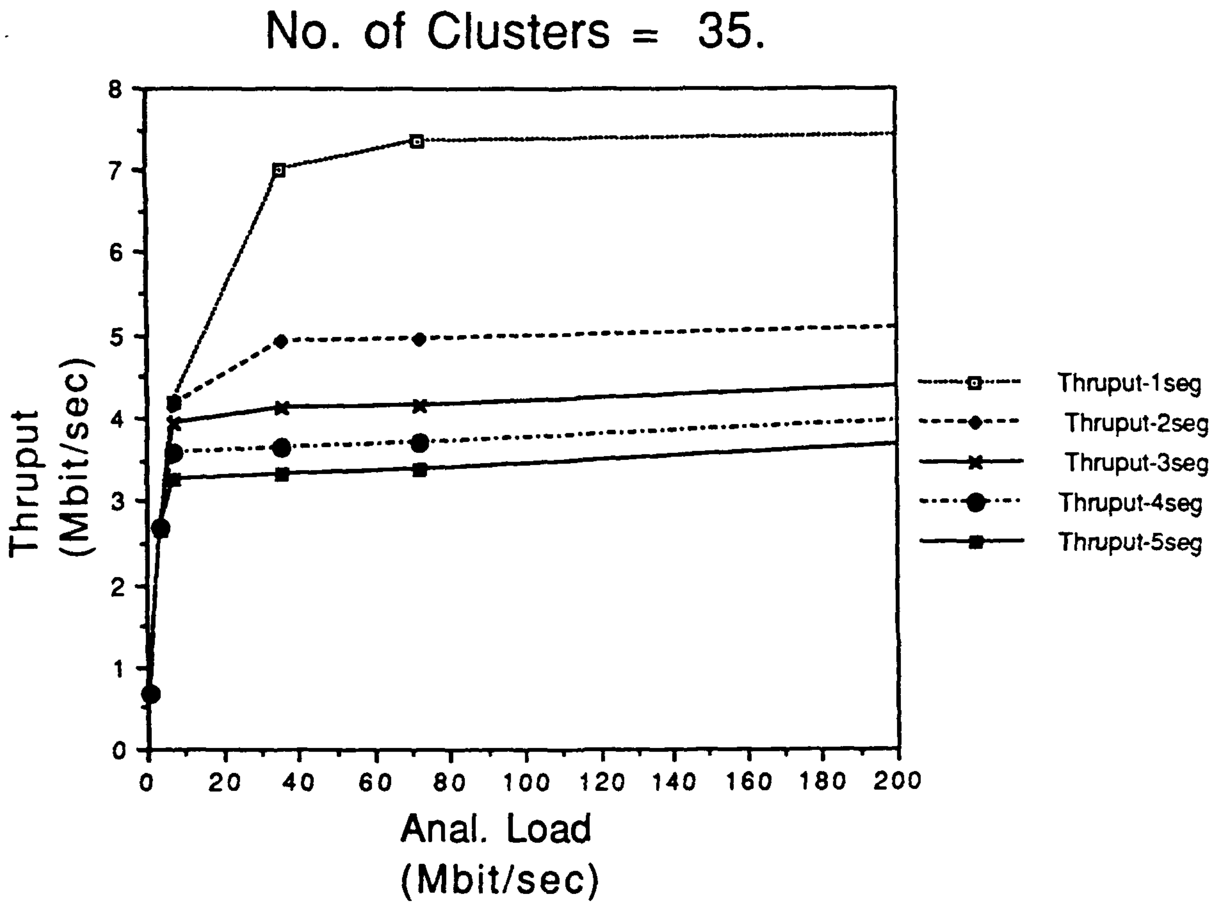


No. of Clusters= 20.
Thruput
(Mbits/sec)
Anal. load
(Mbits/sec)
Thruput-1seg
Thruput-2seg
Thruput-3seg
Thruput-4seg
Thruput-5seg
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200

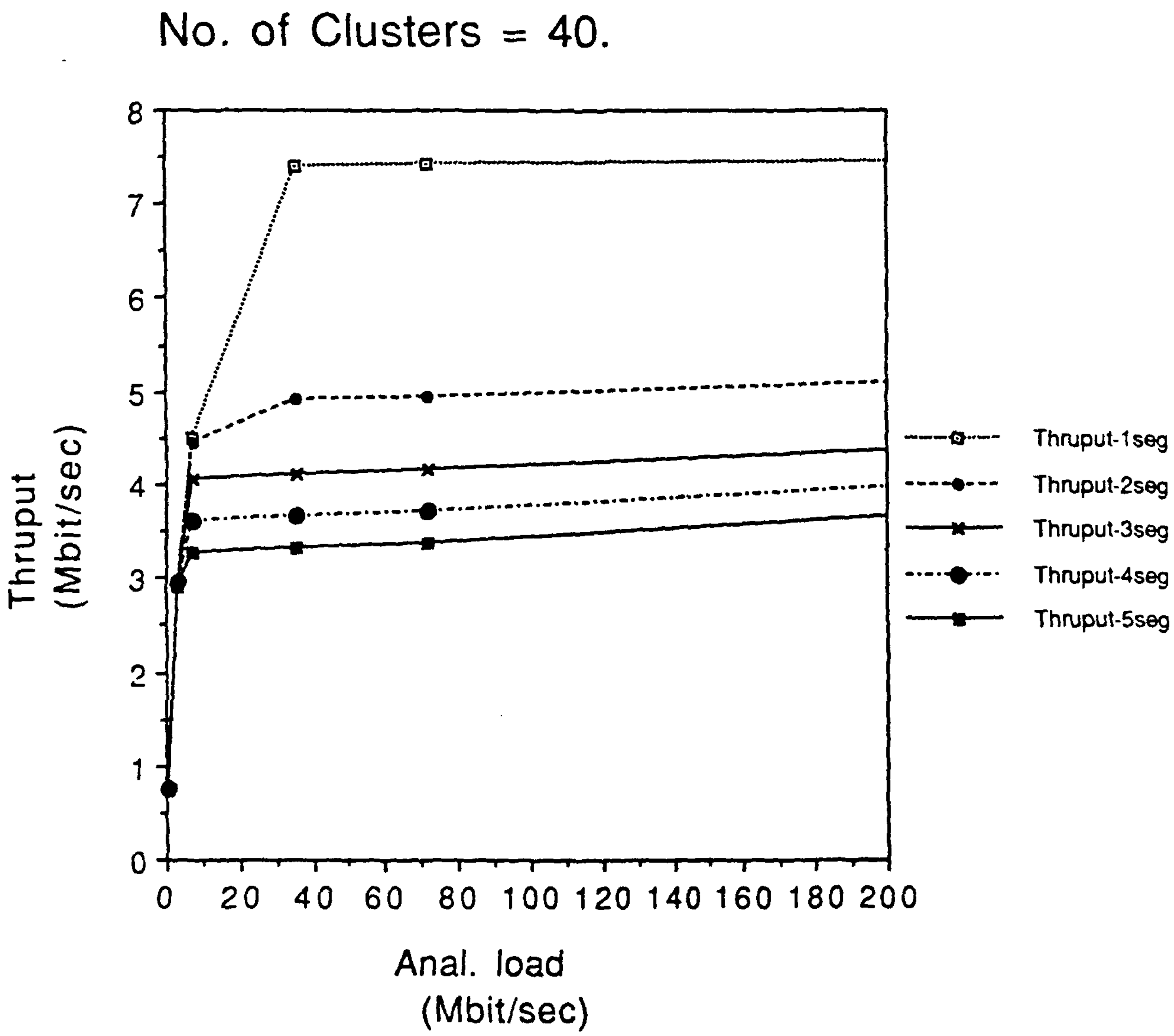
Graph 3


No. of clusters = 30.
Thruput
(Mbit/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbits/sec)
Thruput-1seg
Thruput-2seg
Thruput-3seg
Thruput-4seg
Thruput-5seg

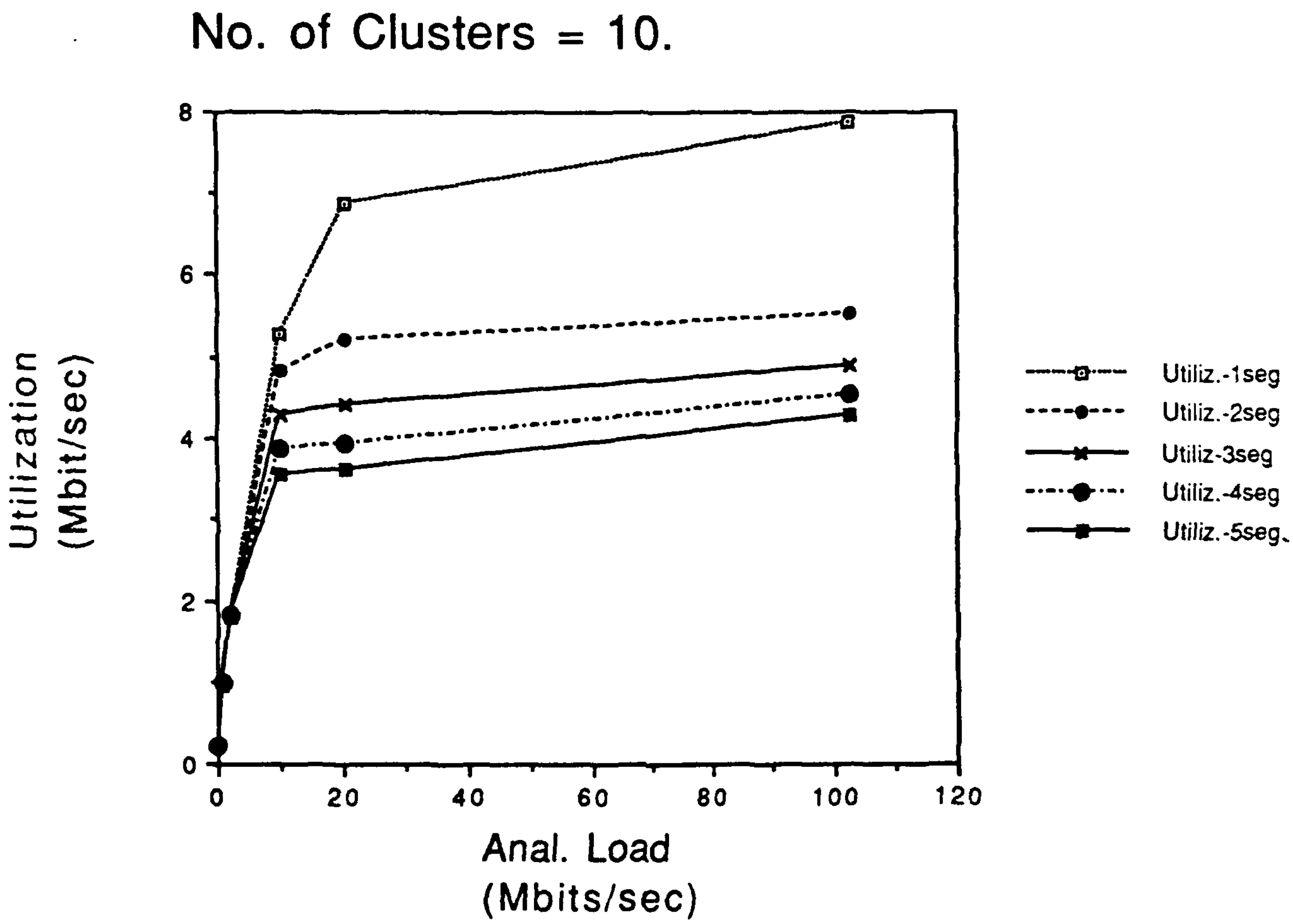
Graph 4


No. of Clusters = 35.
Thruput (Mbit/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. Load (Mbit/sec)
Thruput-1seg
Thruput-2seg
Thruput-3seg
Thruput-4seg
Thruput-5seg

Graph 5

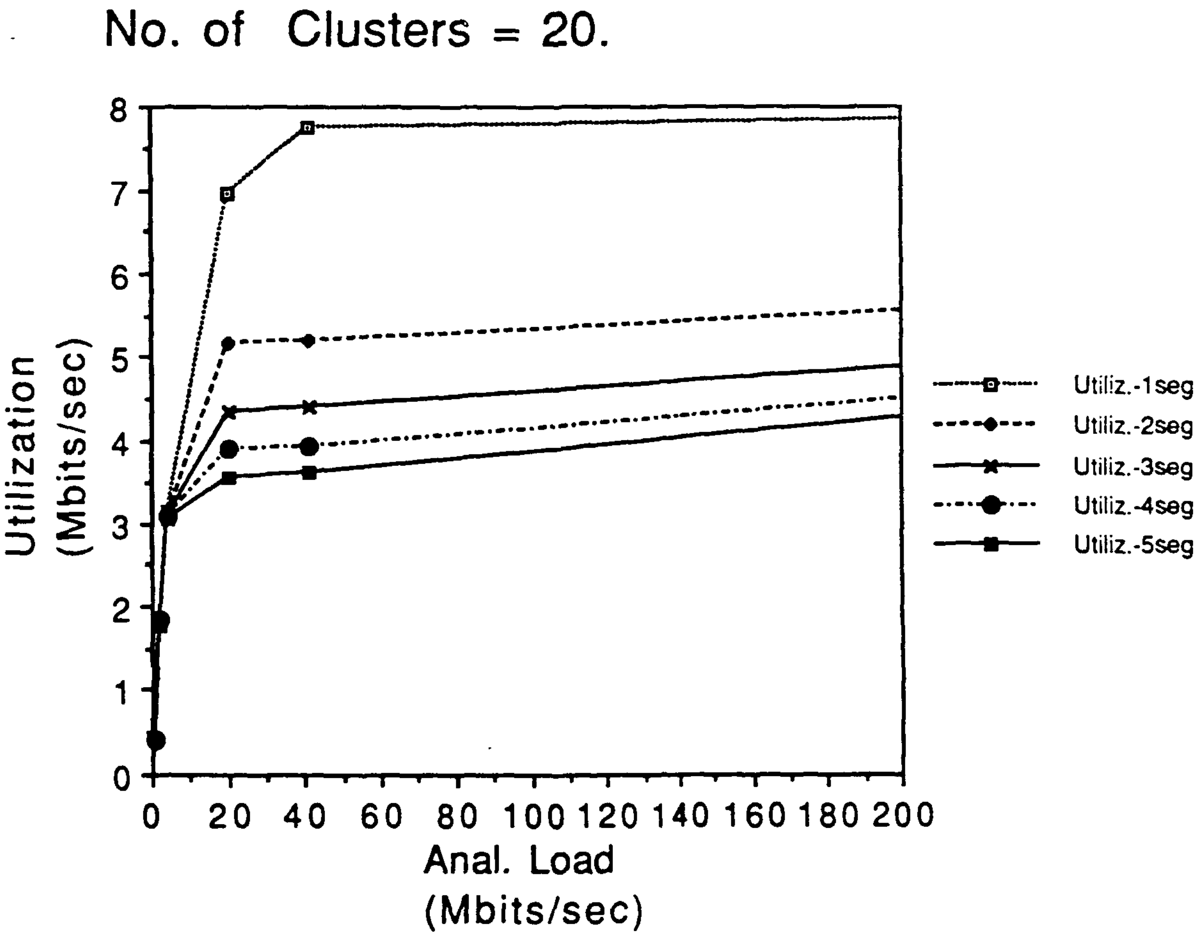

No. of Clusters = 40.
8
7
6
5
4
3
2
1
0
Thruput
(Mbit/sec)
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbit/sec)
Thruput-1seg
Thruput-2seg
Thruput-3seg
Thruput-4seg
Thruput-5seg

Graph 6

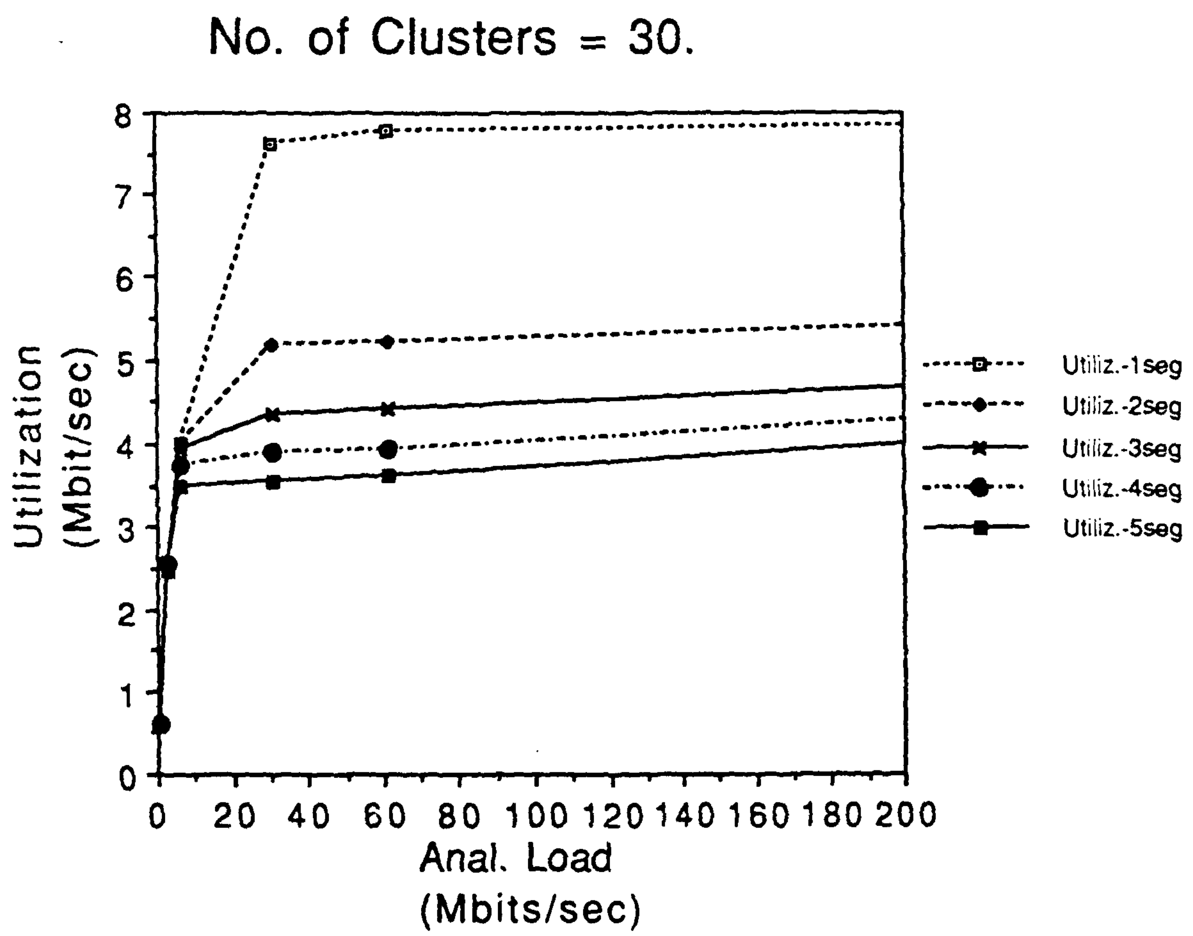


No. of Clusters = 10.
Utilization (Mbit/sec)
8
6
4
2
0
0
20
40
60
80
100
120
Anal. Load (Mbits/sec)
Utiliz.-1seg
Utiliz.-2seg
Utiliz-3seg
Utiliz.-4seg
Utiliz.-5seg

Graph 7

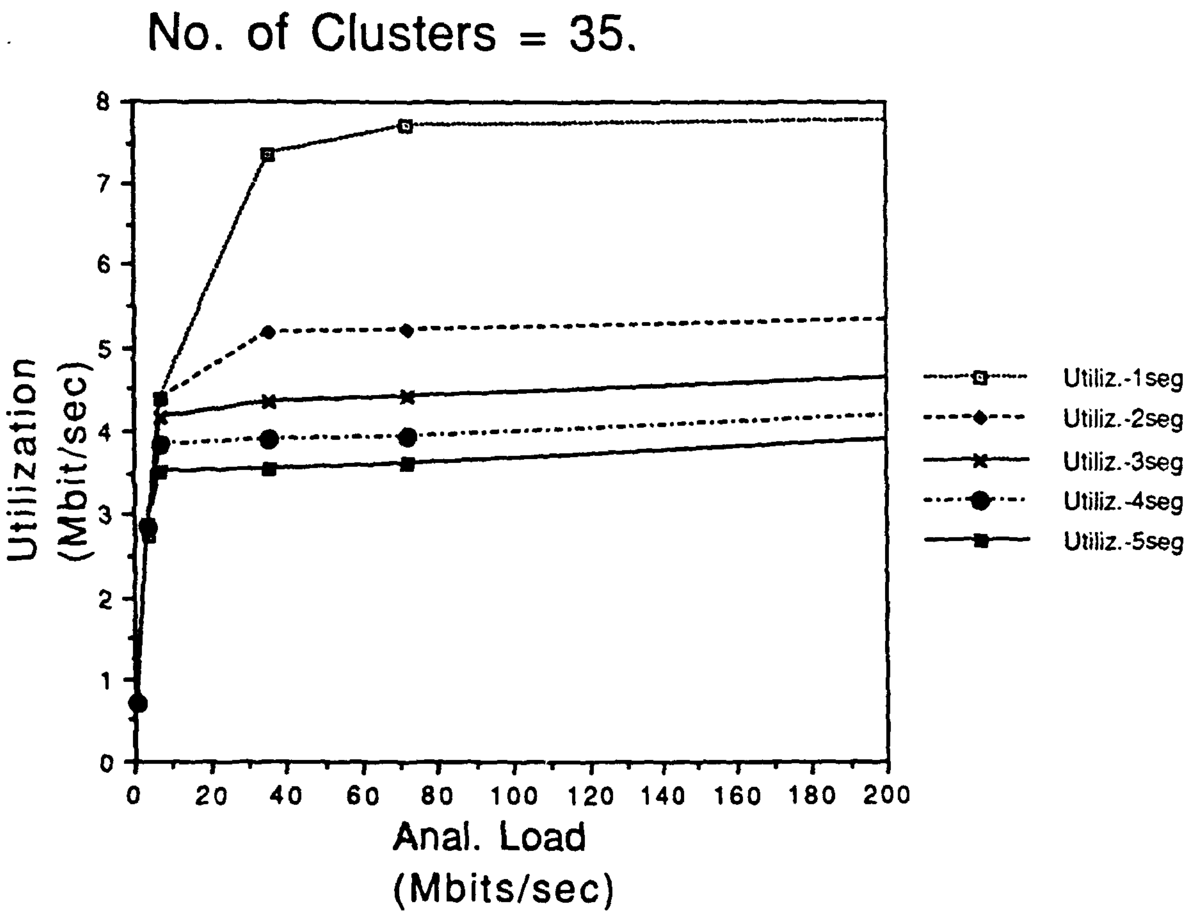

No. of Clusters = 20.
8
7
6
5
4
3
2
1
0
Utilization
(Mbits/sec)
0
20
40
60
80
100
120
140
160
180
200
Anal. Load
(Mbits/sec)
Utiliz.-1seg
Utiliz.-2seg
Utiliz.-3seg
Utiliz.-4seg
Utiliz.-5seg

Graph 8



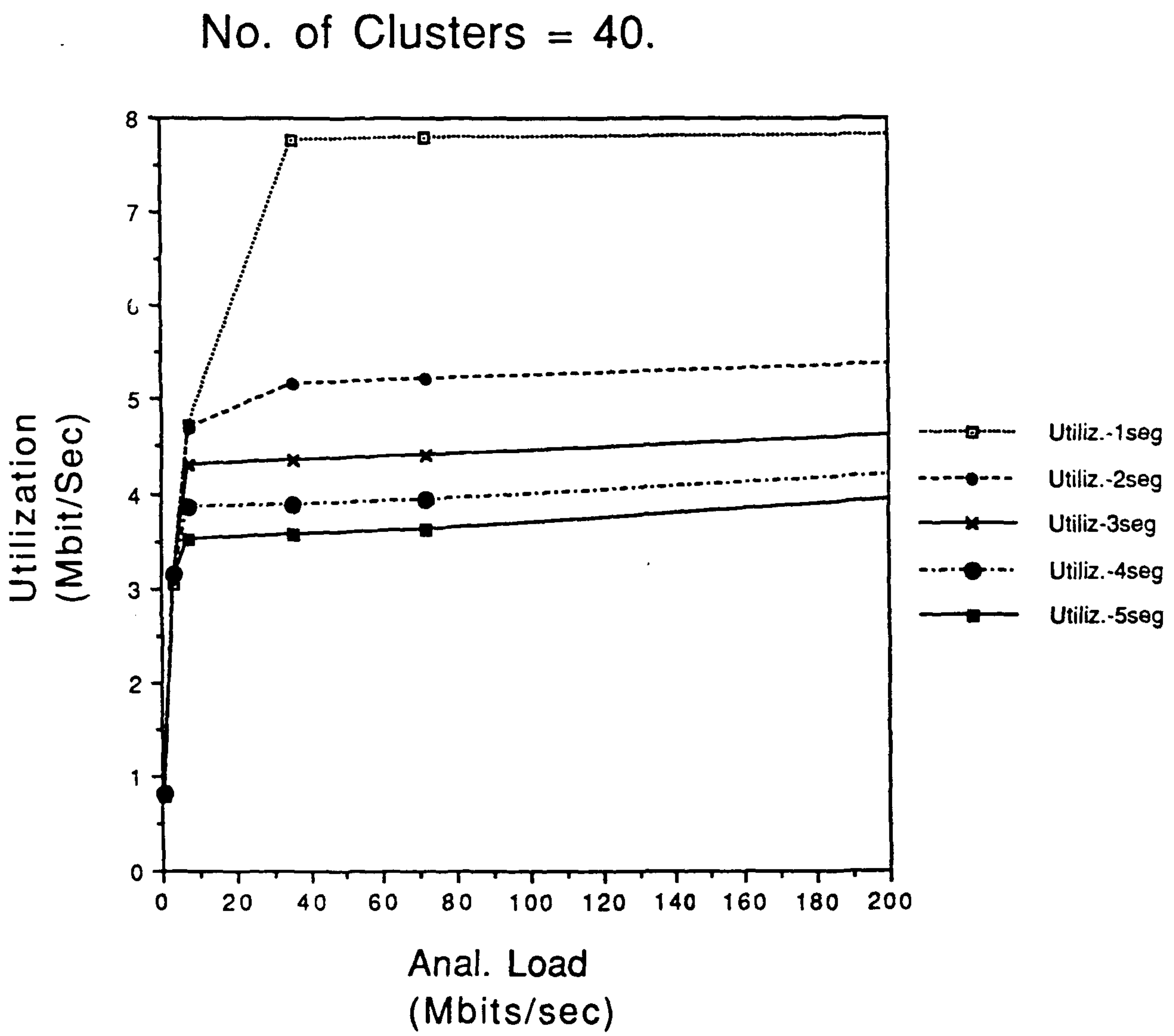
No. of Clusters = 30.
Utilization (Mbit/sec)
0
1
2
3
4
5
6
7
8
0 20 40 60 80 100 120 140 160 180 200
Anal. Load (Mbits/sec)
Utiliz.-1seg
Utiliz.-2seg
Utiliz.-3seg
Utiliz.-4seg
Utiliz.-5seg

Graph 9

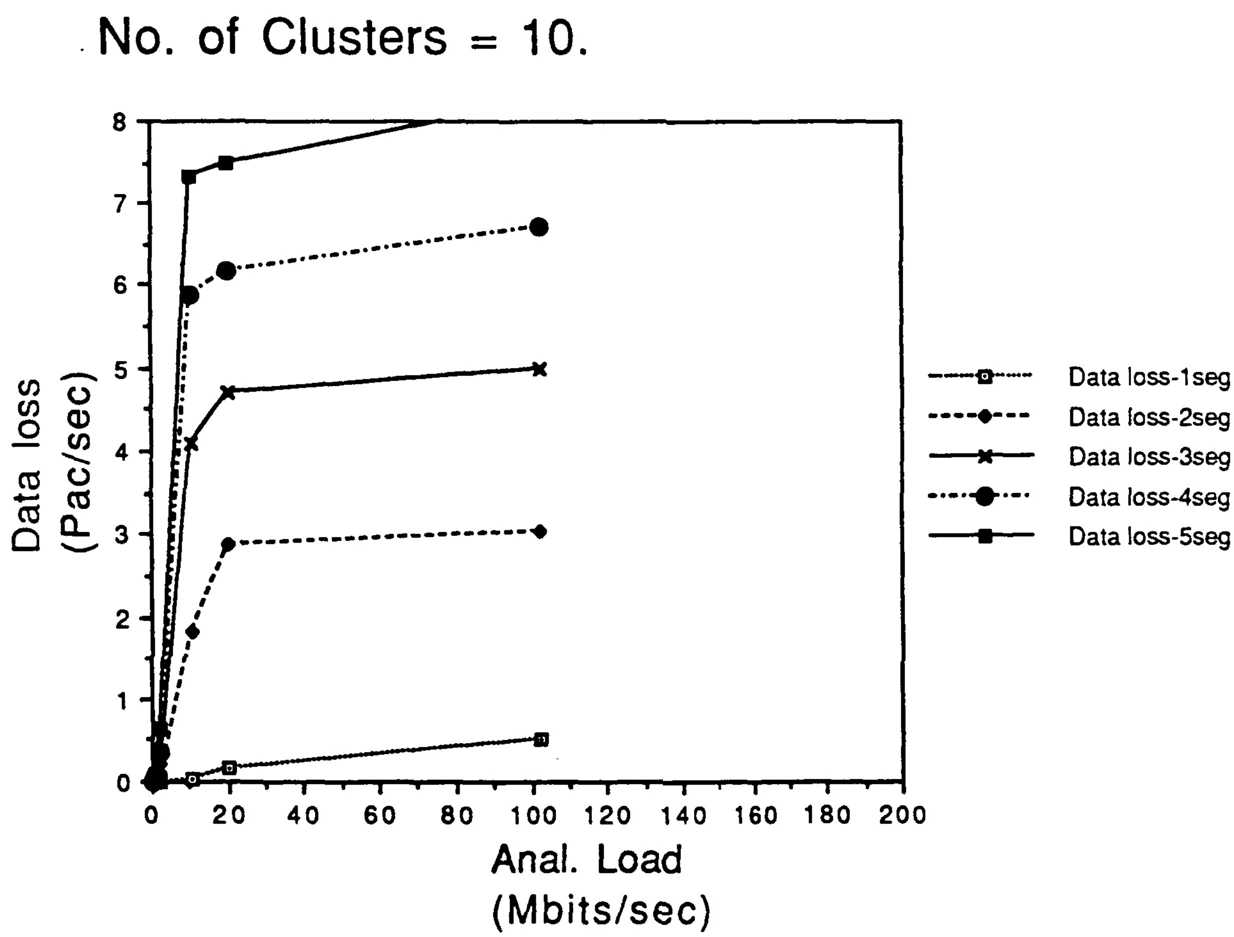


No. of Clusters = 35.
Utilization (Mbit/sec)
Anal. Load (Mbits/sec)
Utiliz.-1seg
Utiliz.-2seg
Utiliz.-3seg
Utiliz.-4seg
Utiliz.-5seg
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200

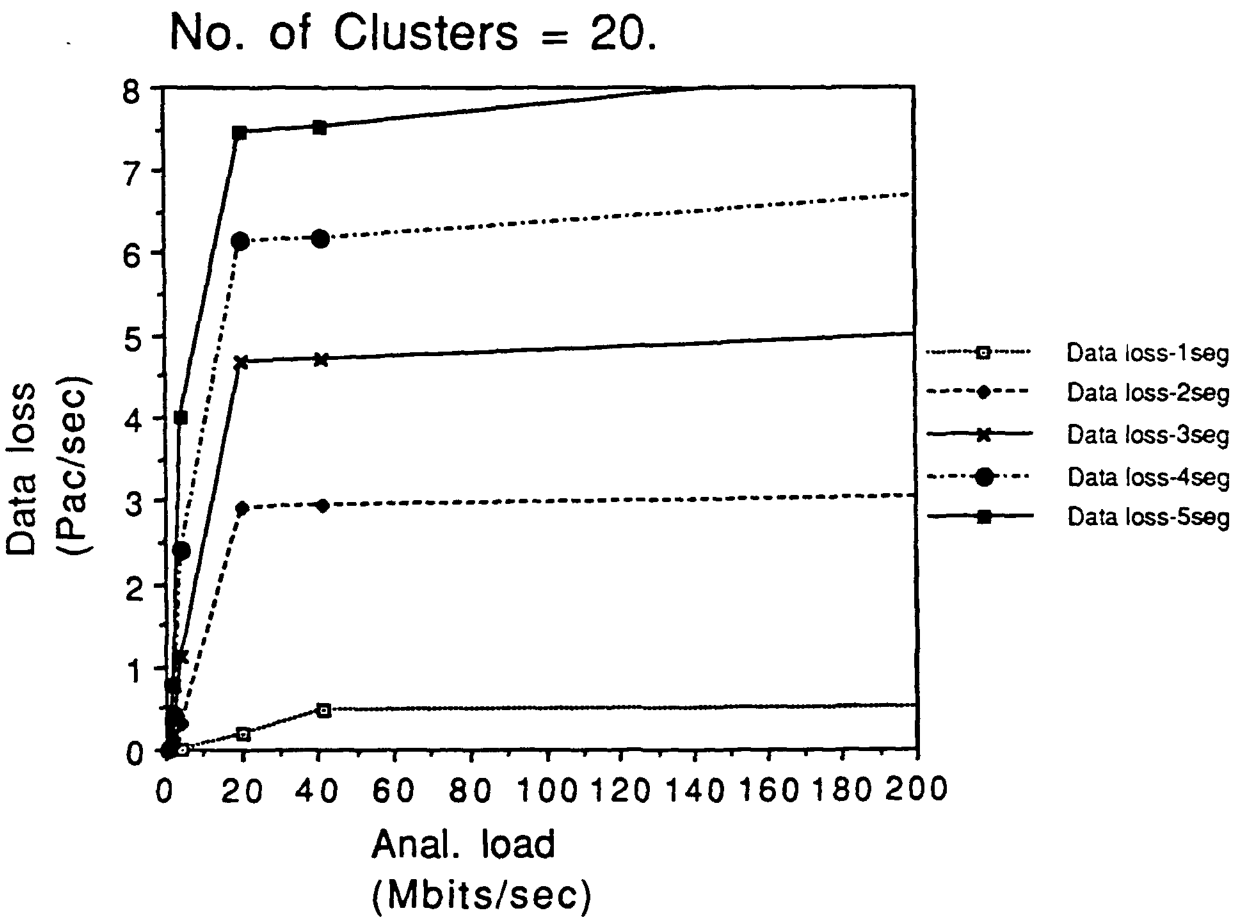
Graph 10


No. of Clusters = 40.
Utilization
(Mbit/Sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. Load
(Mbits/sec)
Utiliz.-1seg
Utiliz.-2seg
Utiliz-3seg
Utiliz.-4seg
Utiliz.-5seg

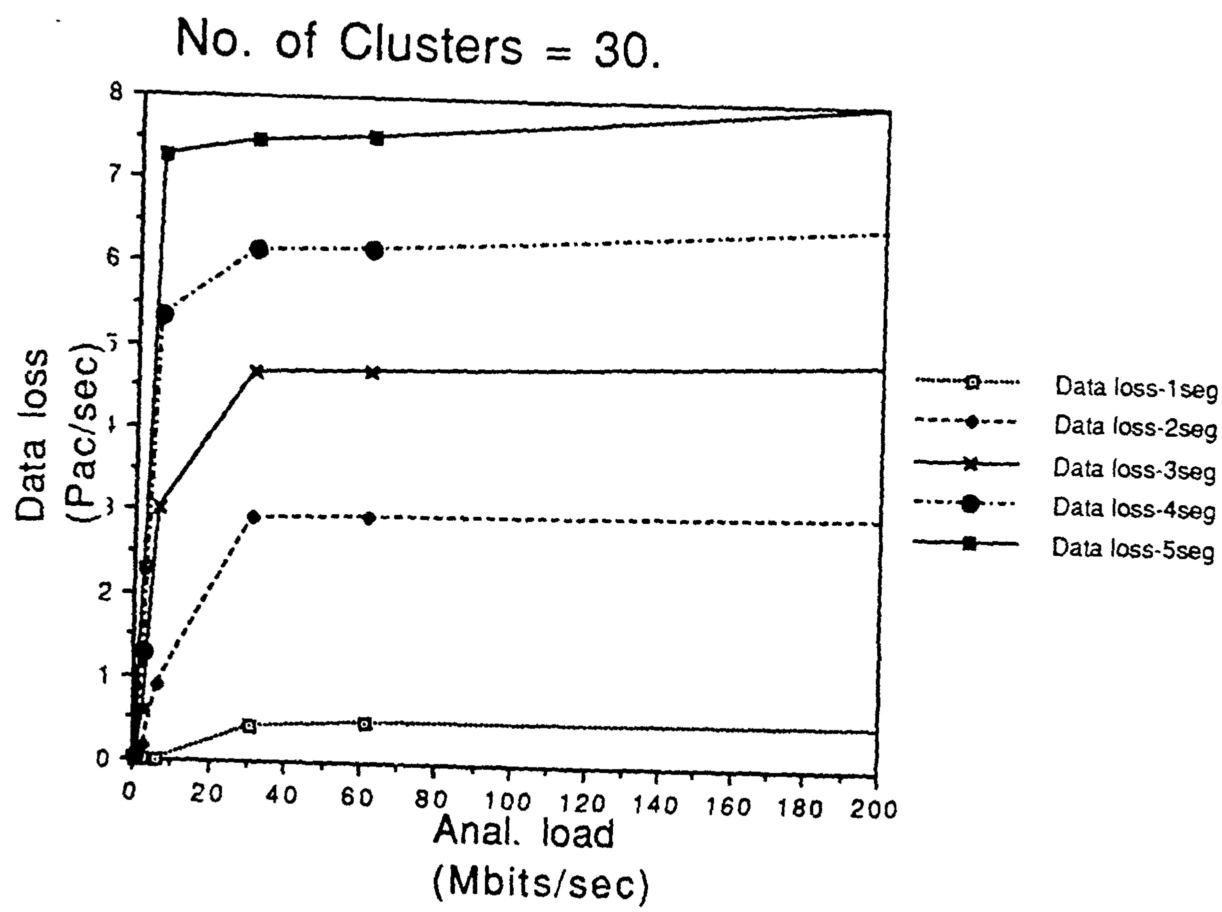
Graph 11


No. of Clusters = 10.
Data loss
(Pac/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. Load
(Mbits/sec)
Data loss-1seg
Data loss-2seg
Data loss-3seg
Data loss-4seg
Data loss-5seg

Graph 12

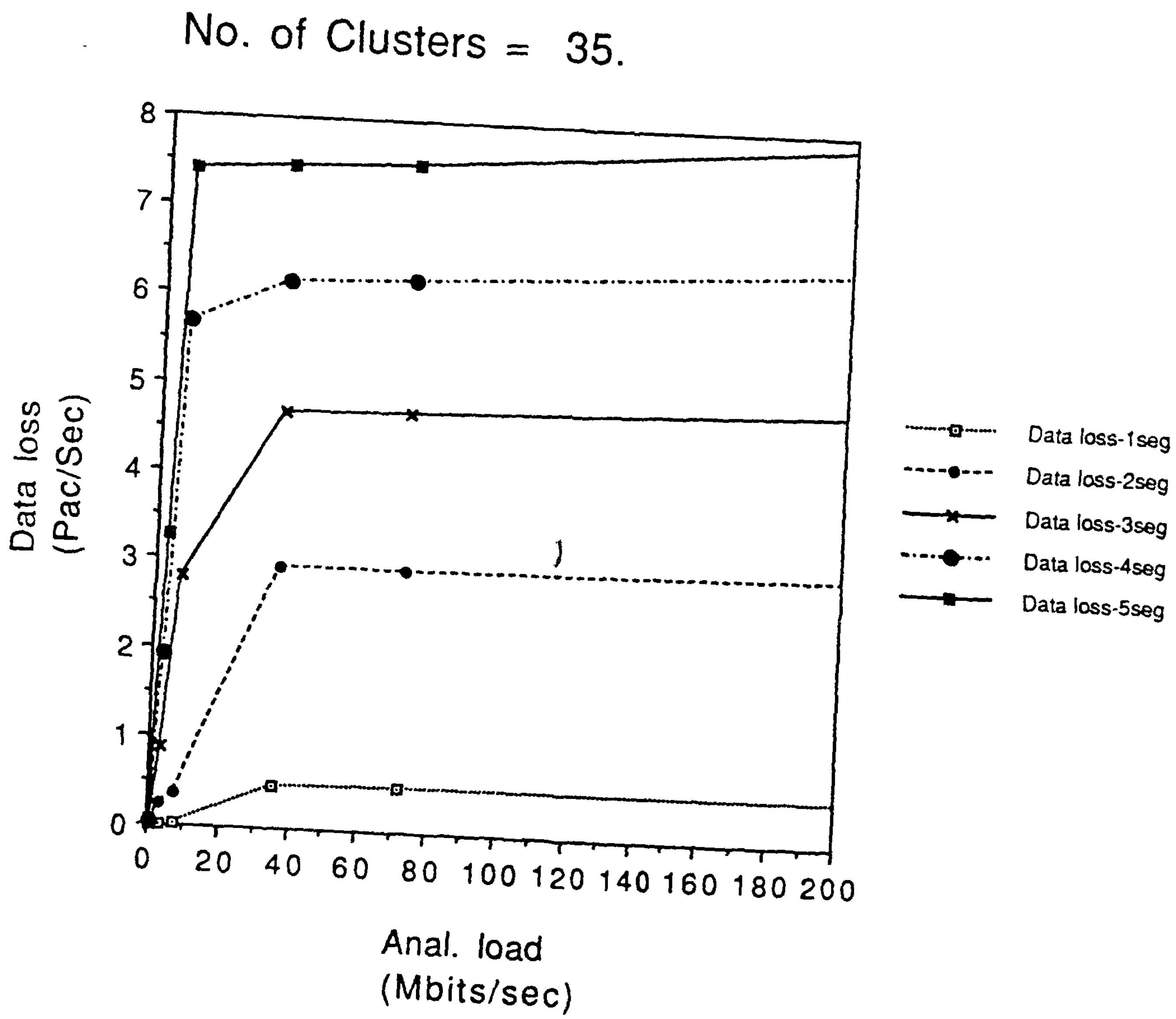

No. of Clusters = 20.
Data loss
(Pac/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbits/sec)
Data loss-1seg
Data loss-2seg
Data loss-3seg
Data loss-4seg
Data loss-5seg

Graph 13

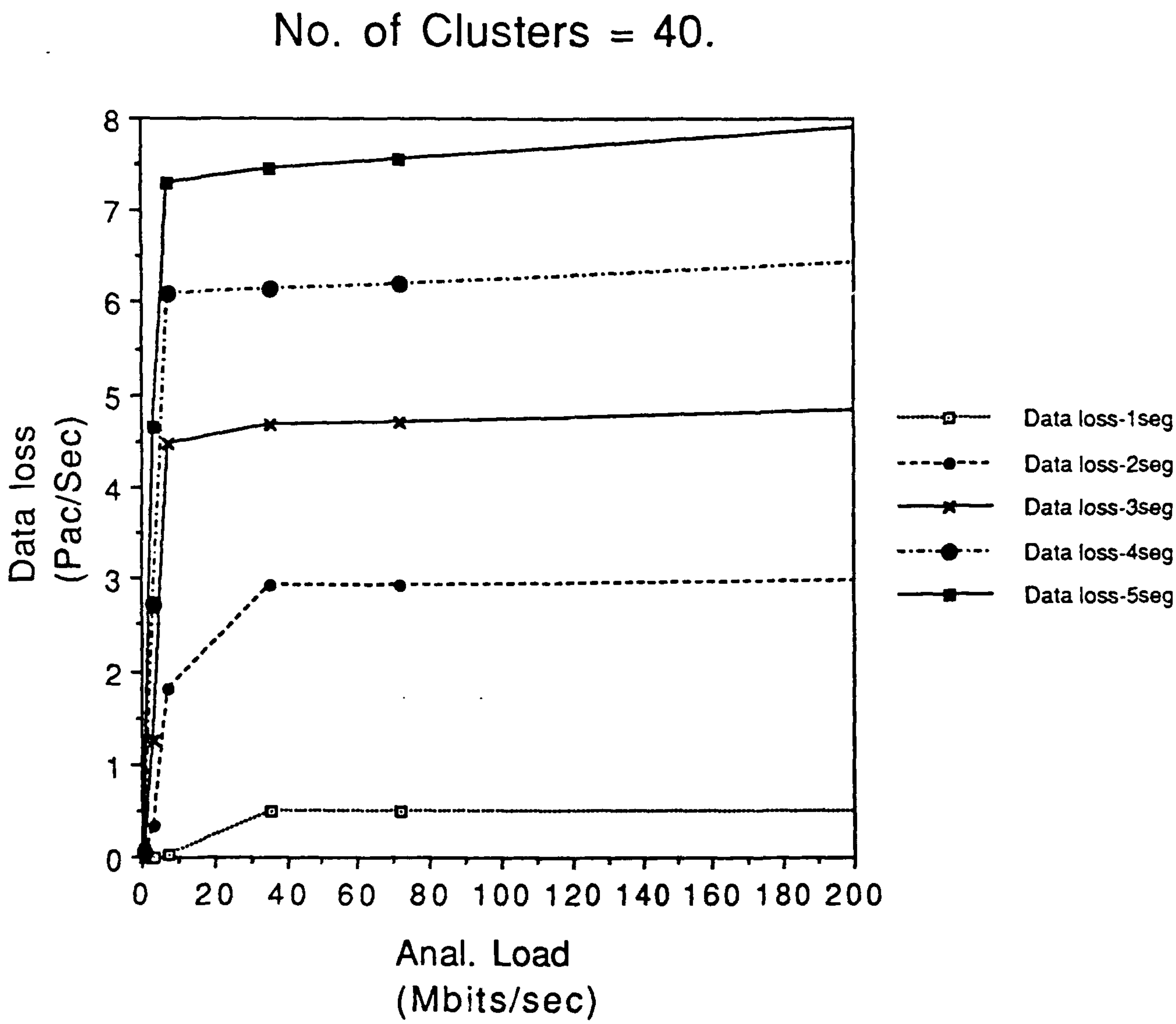


No. of Clusters = 30.
Data loss
(Pac/sec)
Anal. load
(Mbits/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Data loss-1seg
Data loss-2seg
Data loss-3seg
Data loss-4seg
Data loss-5seg

Graph 14

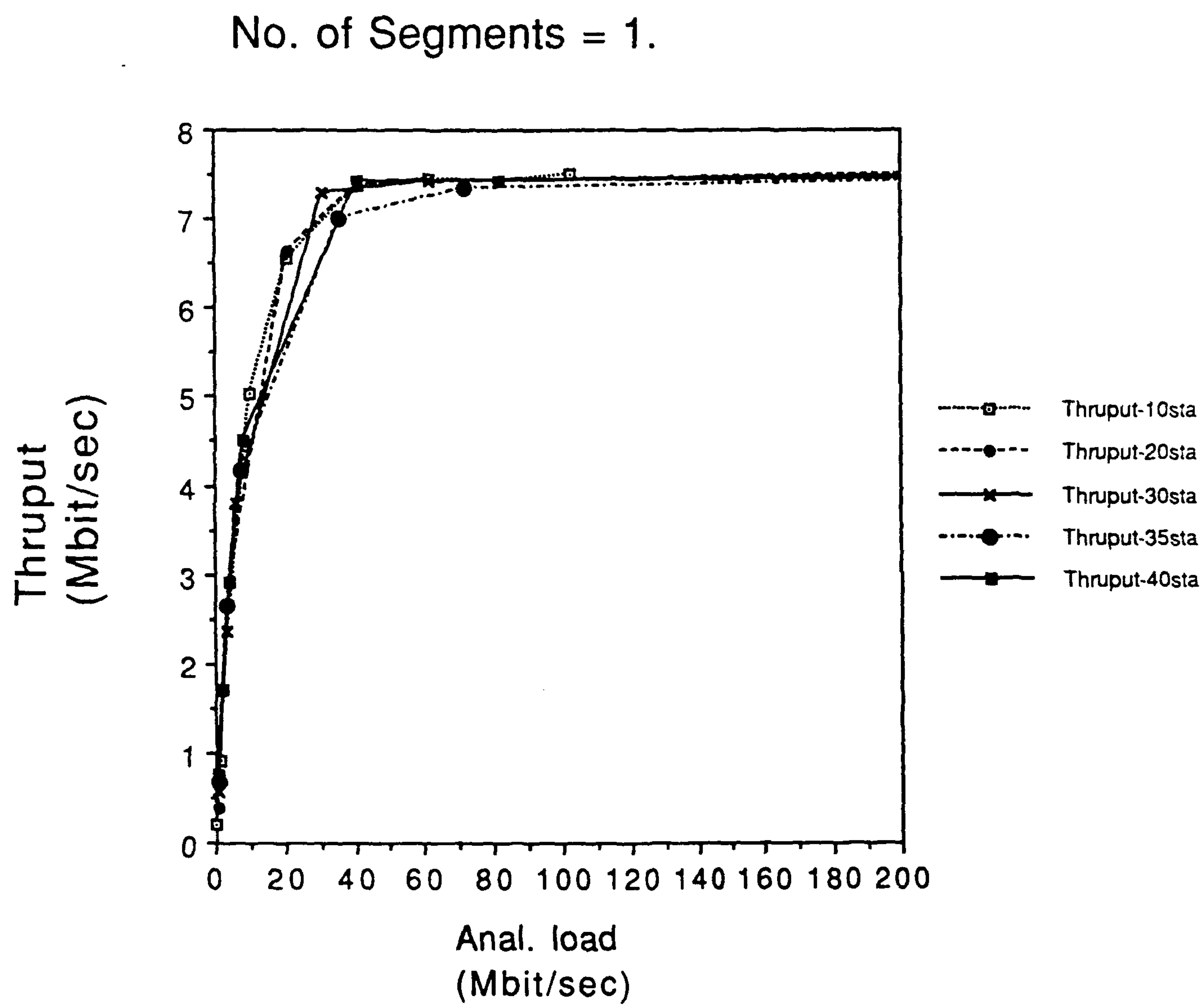

No. of Clusters = 35.
Data loss
(Pac/Sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbits/sec)
Data loss-1seg
Data loss-2seg
Data loss-3seg
Data loss-4seg
Data loss-5seg

Graph 15

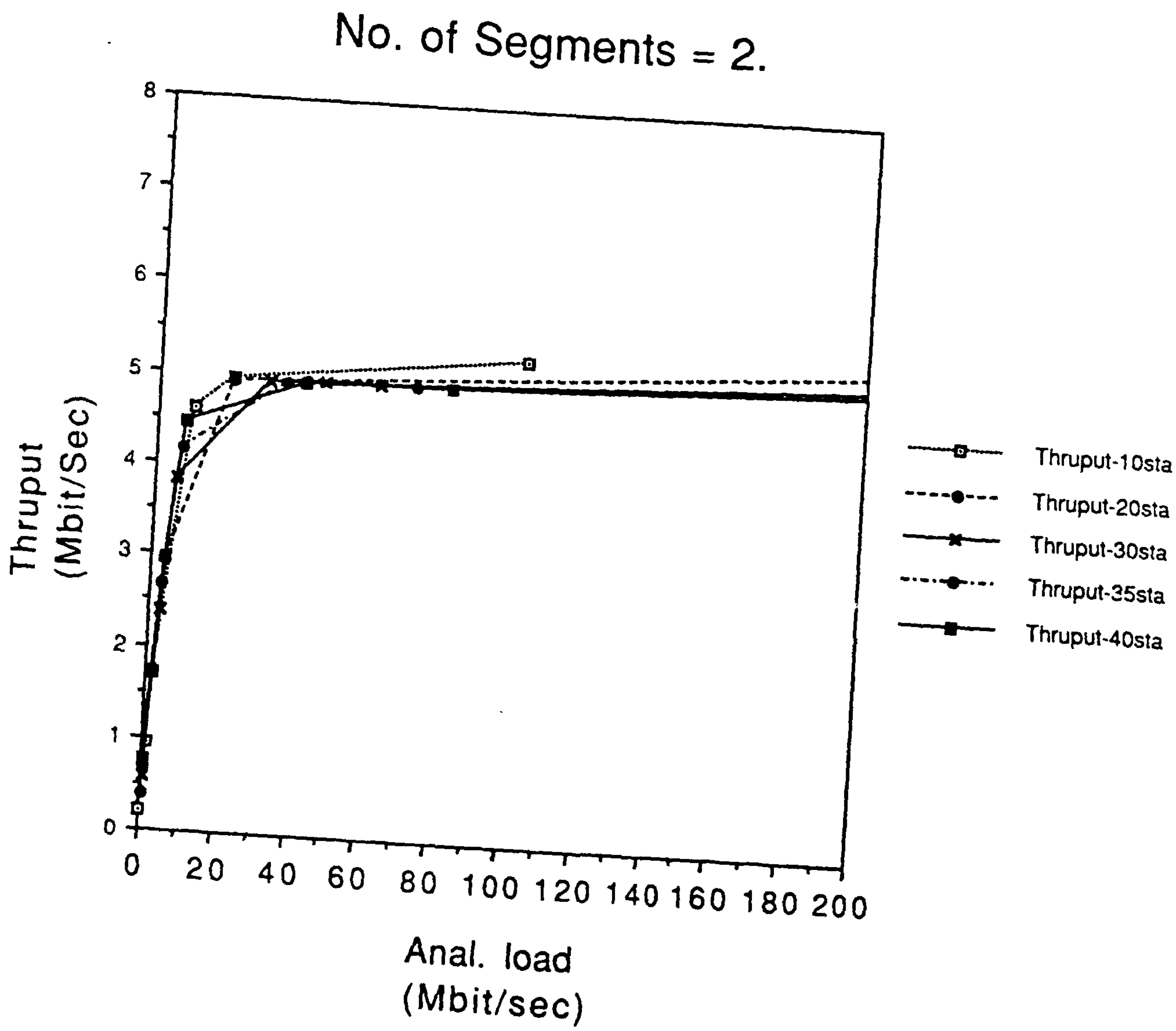


No. of Clusters = 40.
Data loss
(Pac/Sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. Load
(Mbits/sec)
Data loss-1seg
Data loss-2seg
Data loss-3seg
Data loss-4seg
Data loss-5seg

Graph 16

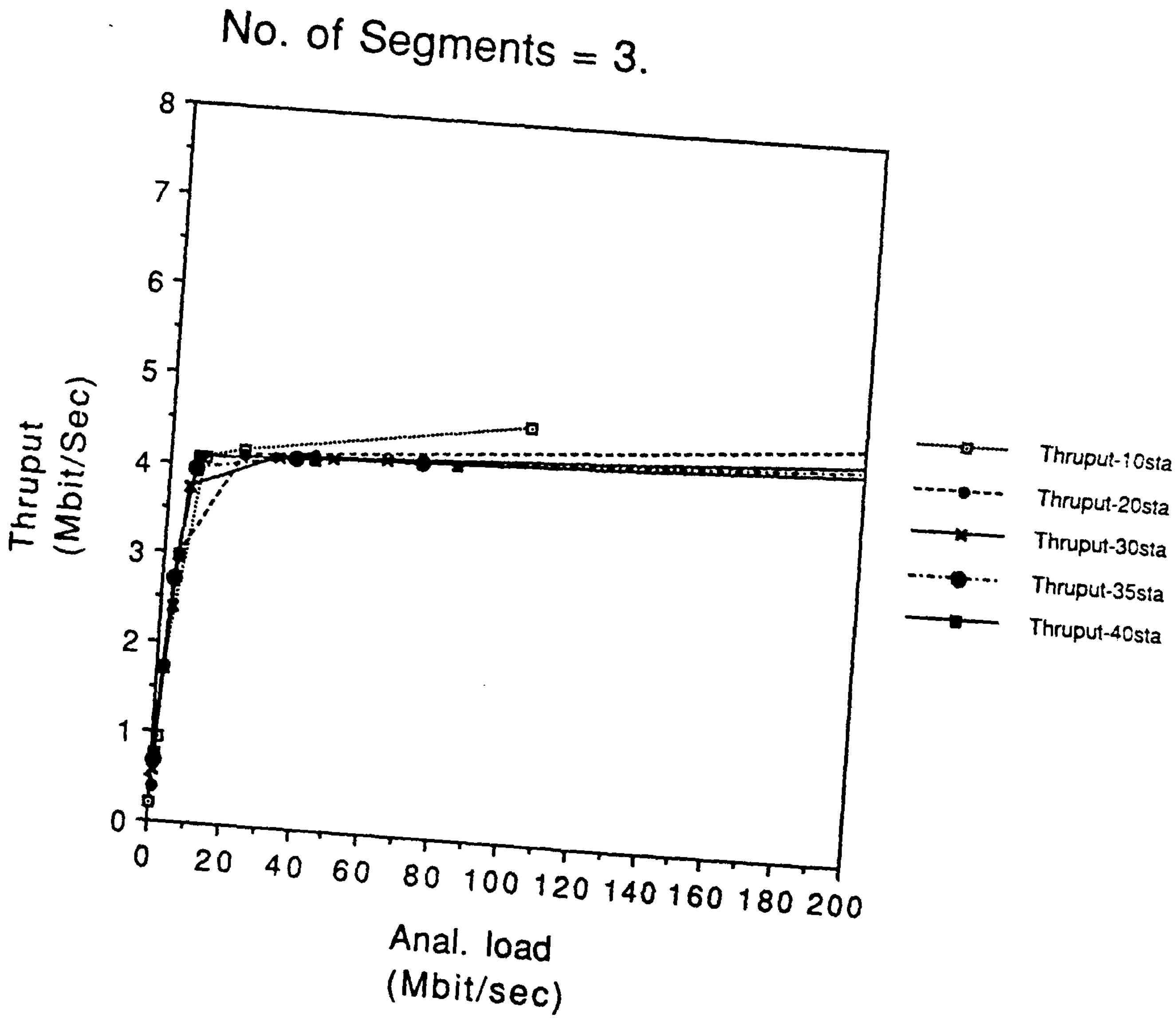

No. of Segments = 1.
8
7
6
5
4
3
2
1
0
0
20
40
60
80
100
120
140
160
180
200
Thruput
(Mbit/sec)
Anal. load
(Mbit/sec)
Thruput-10sta
Thruput-20sta
Thruput-30sta
Thruput-35sta
Thruput-40sta

Graph 17

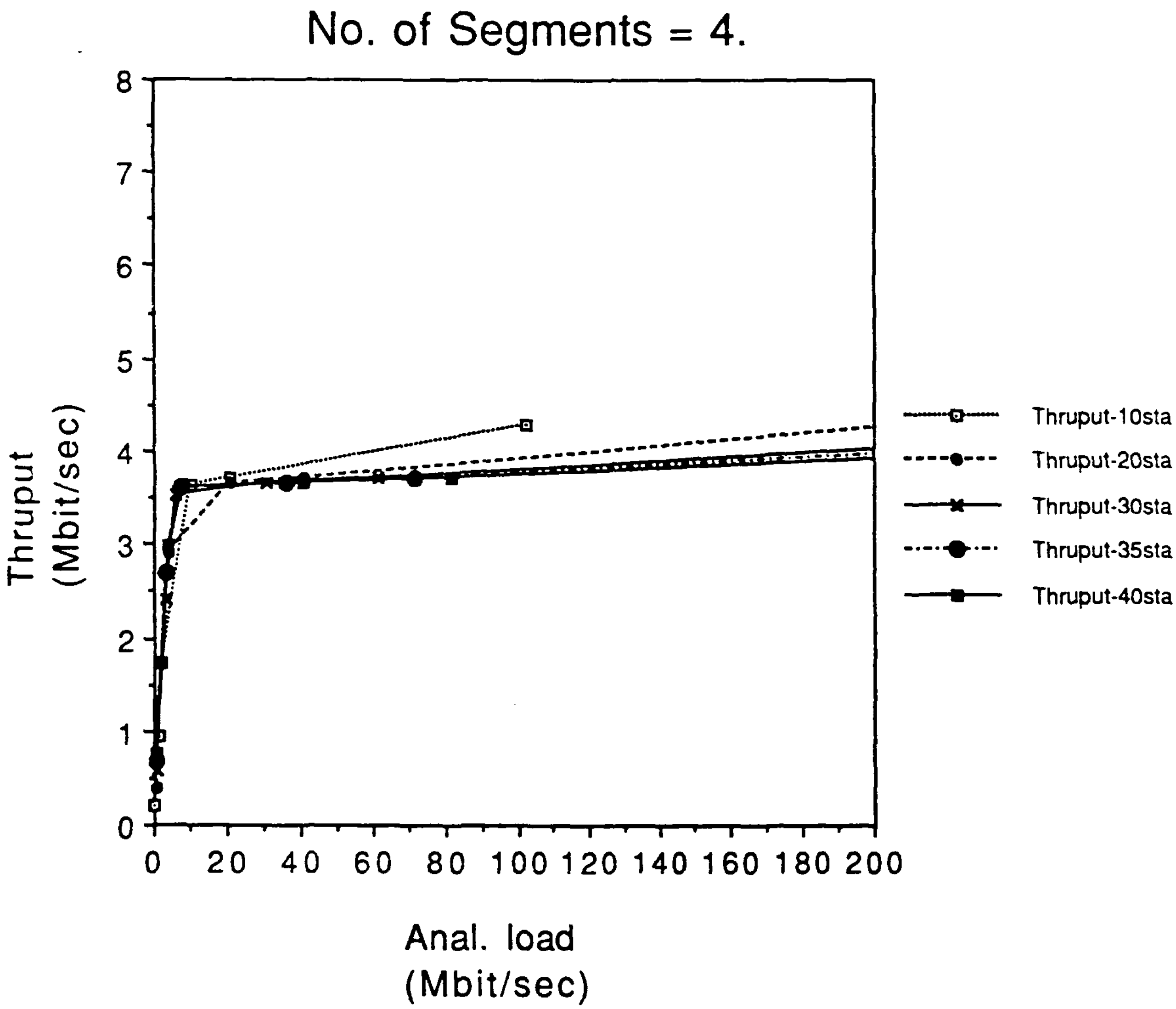


No. of Segments = 2.
8
7
6
5
4
3
2
1
0
Thruput
(Mbit/Sec)
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbit/sec)
Thruput-10sta
Thruput-20sta
Thruput-30sta
Thruput-35sta
Thruput-40sta

Graph 18

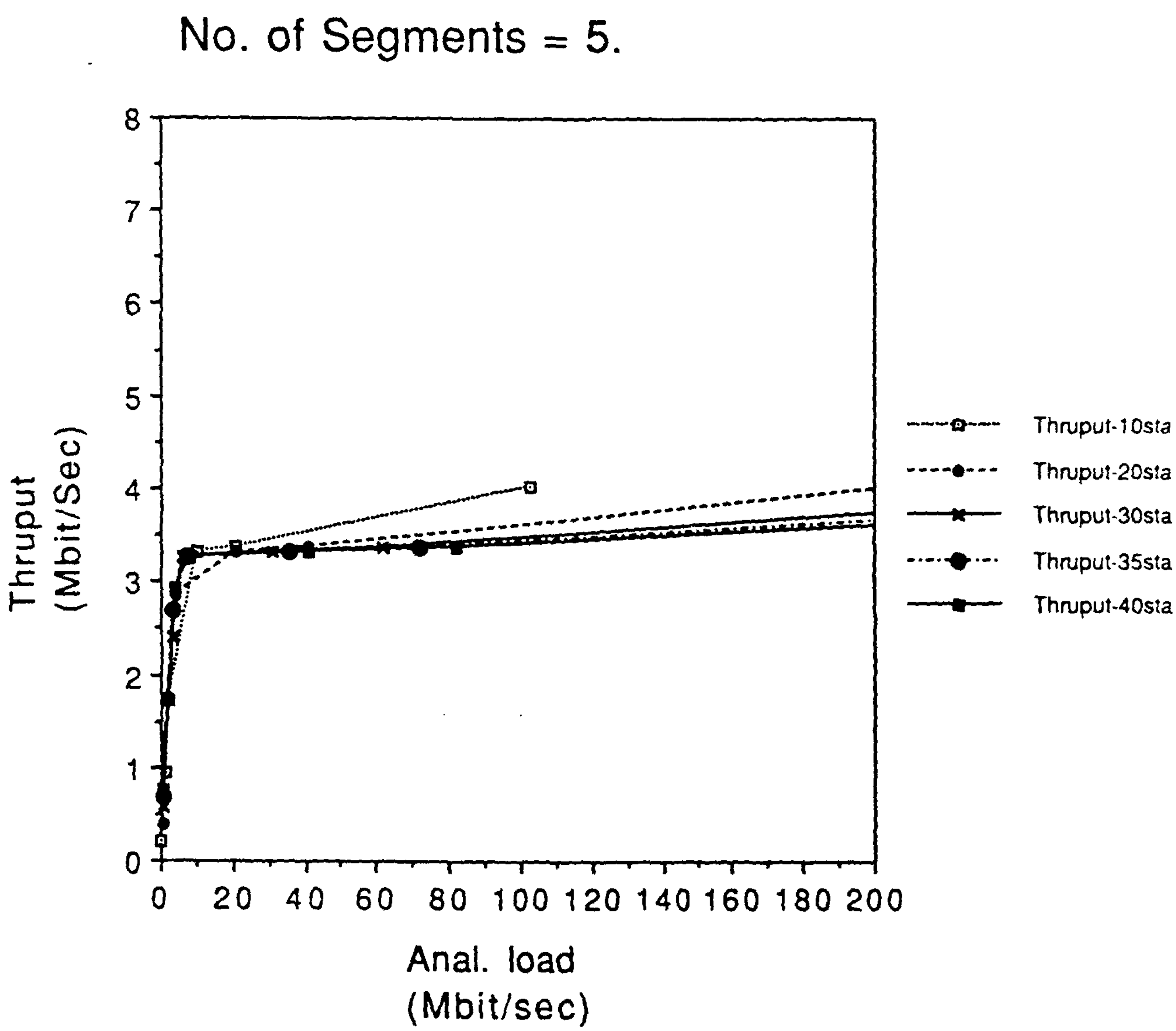

No. of Segments = 3.
Thruput
(Mbit/Sec)
8
7
6
5
4
3
2
1
0
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbit/sec)
Thruput-10sta
Thruput-20sta
Thruput-30sta
Thruput-35sta
Thruput-40sta

Graph 19



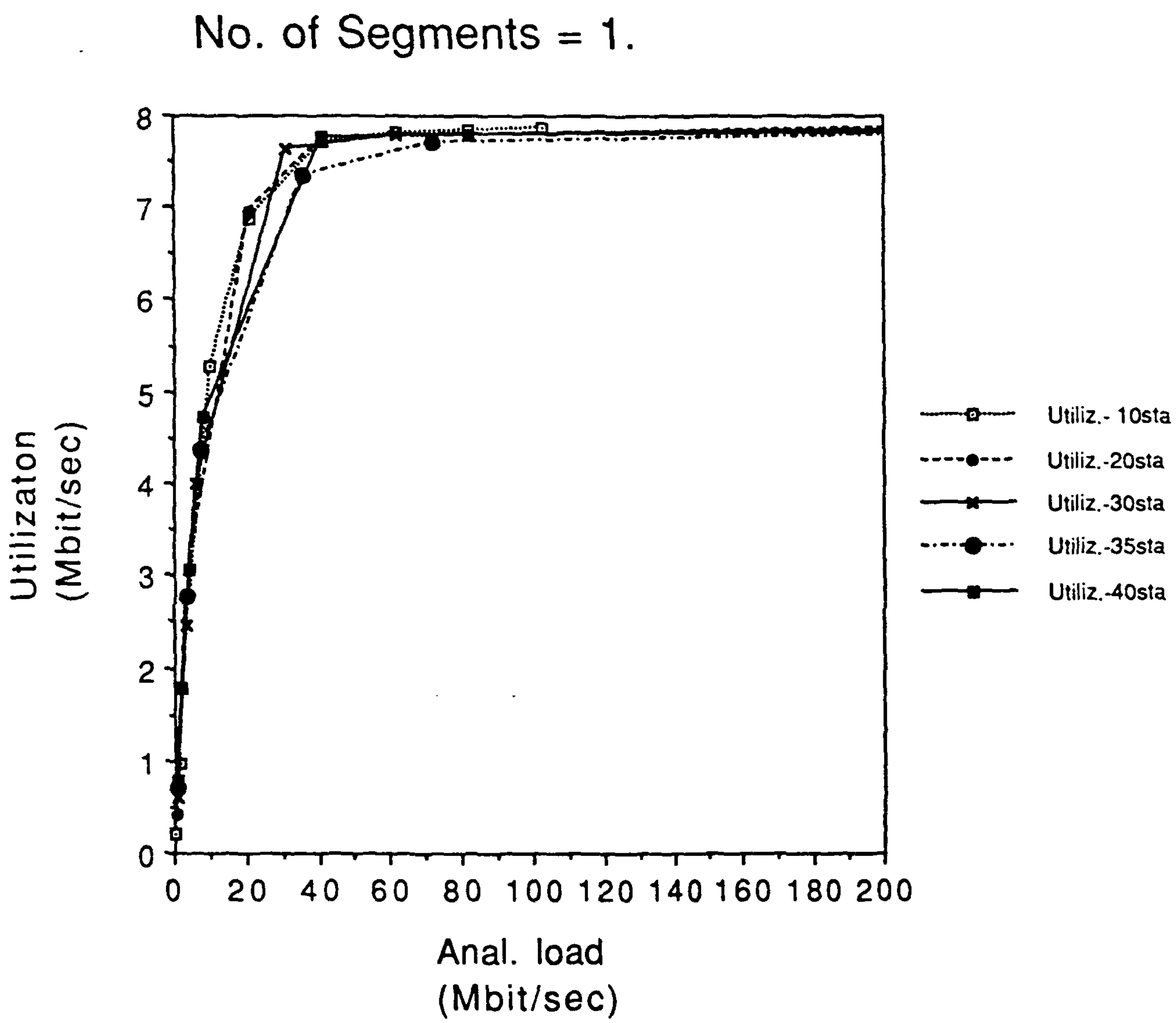
No. of Segments = 4.
Thruput (Mbit/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. load (Mbit/sec)
Thruput-10sta
Thruput-20sta
Thruput-30sta
Thruput-35sta
Thruput-40sta

Graph 20



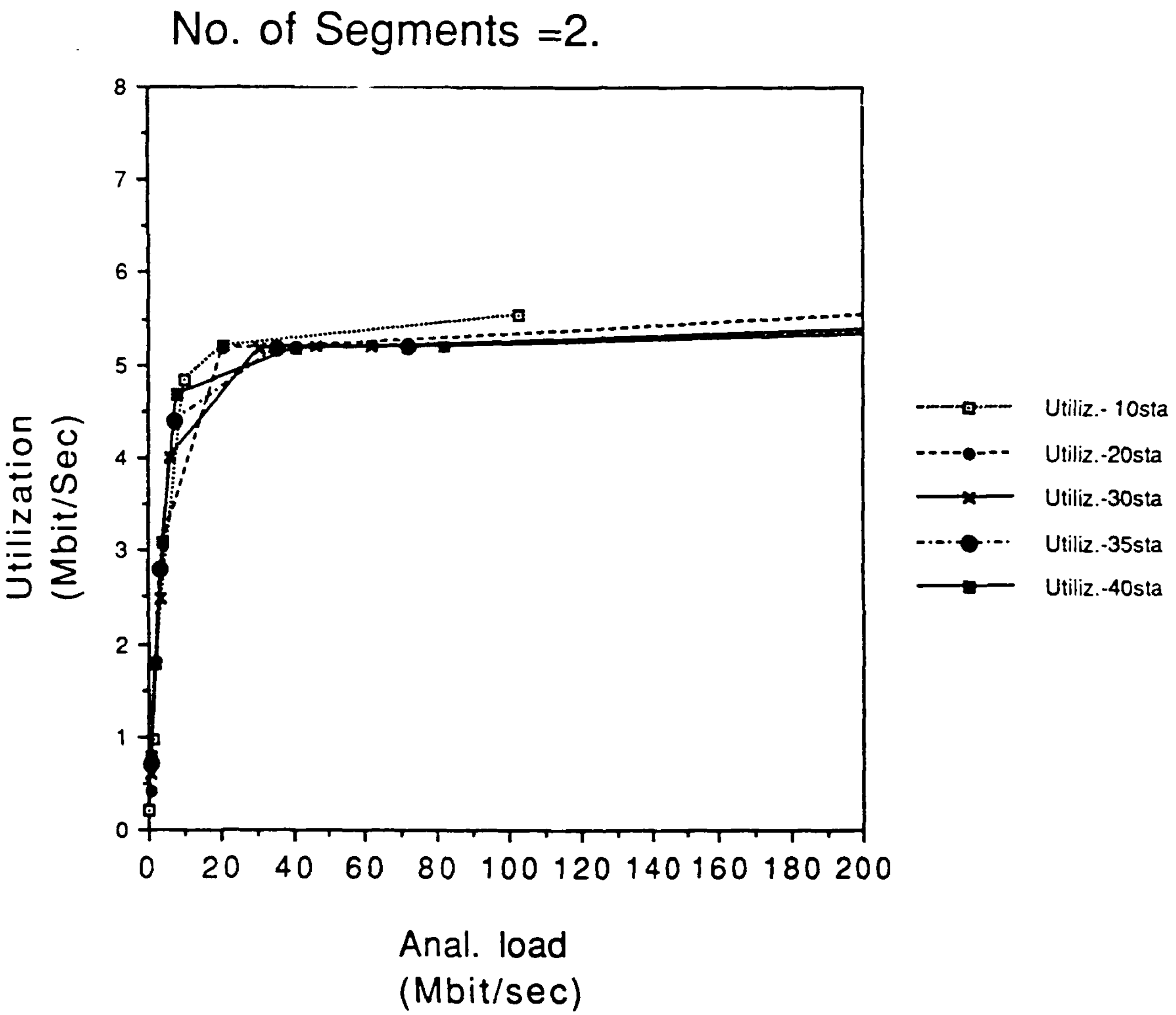
No. of Segments = 5.
Thruput
(Mbit/Sec)
Anal. load
(Mbit/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Thruput-10sta
Thruput-20sta
Thruput-30sta
Thruput-35sta
Thruput-40sta

Graph 21

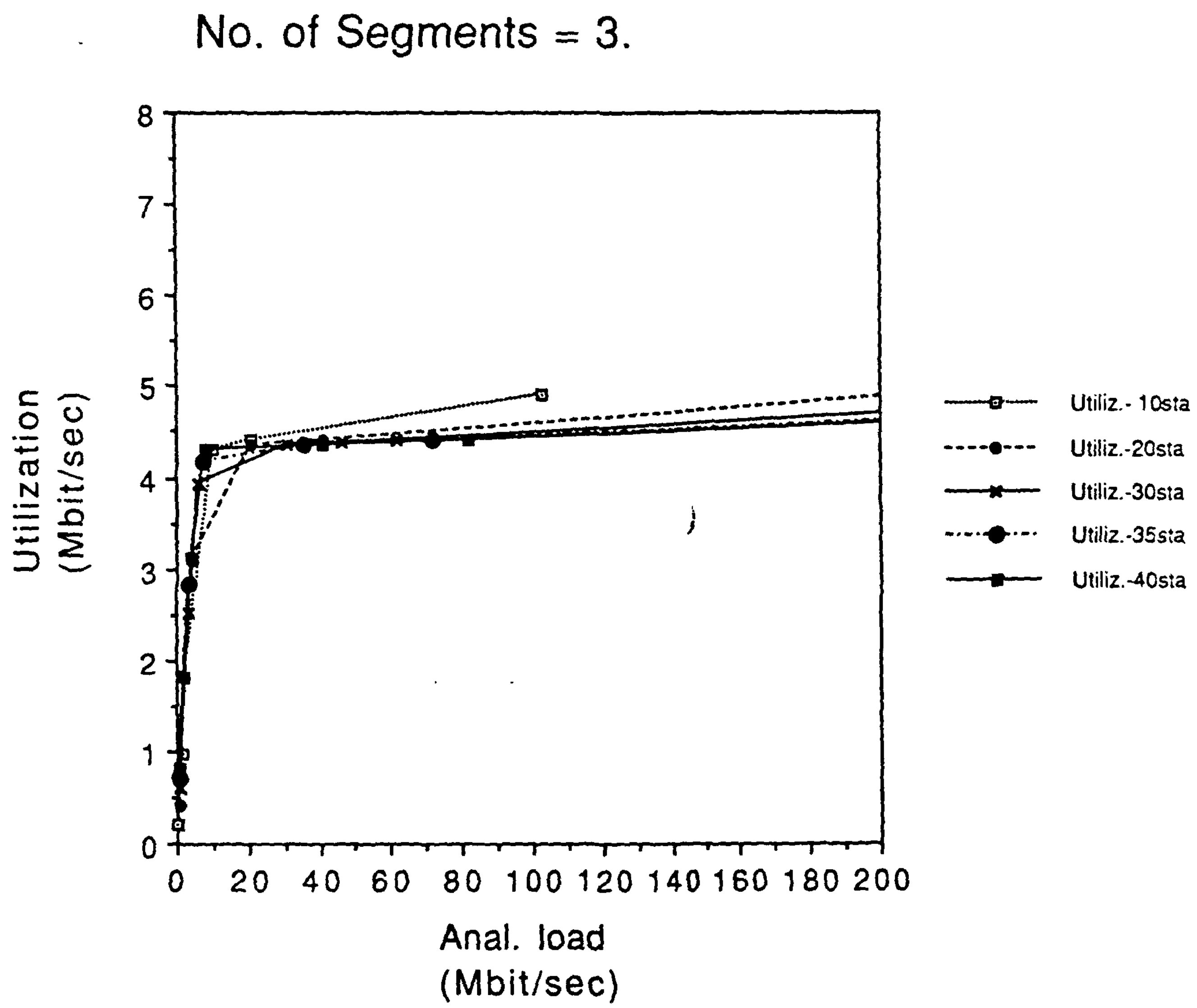


No. of Segments = 1.
Utilizaton (Mbit/sec)
0
1
2
3
4
5
6
7
8
0 20 40 60 80 100 120 140 160 180 200
Anal. load (Mbit/sec)
Utiliz.- 10sta
Utiliz.-20sta
Utiliz.-30sta
Utiliz.-35sta
Utiliz.-40sta

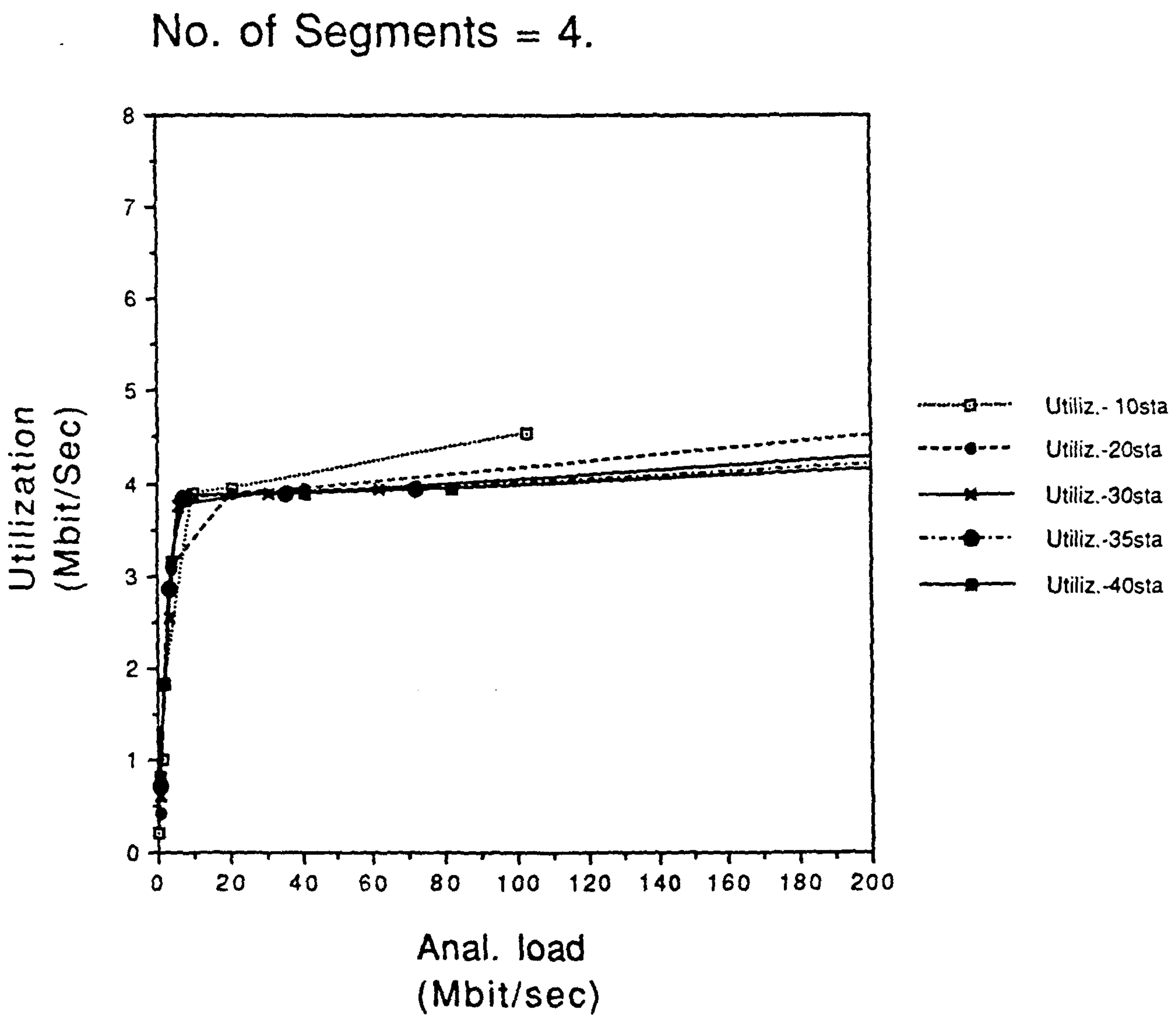
Graph 22


No. of Segments =2.
Utilization (Mbit/Sec)
Anal. load (Mbit/sec)
0
1
2
3
4
5
6
7
8
0 20 40 60 80 100 120 140 160 180 200
Utiliz.- 10sta
Utiliz.-20sta
Utiliz.-30sta
Utiliz.-35sta
Utiliz.-40sta

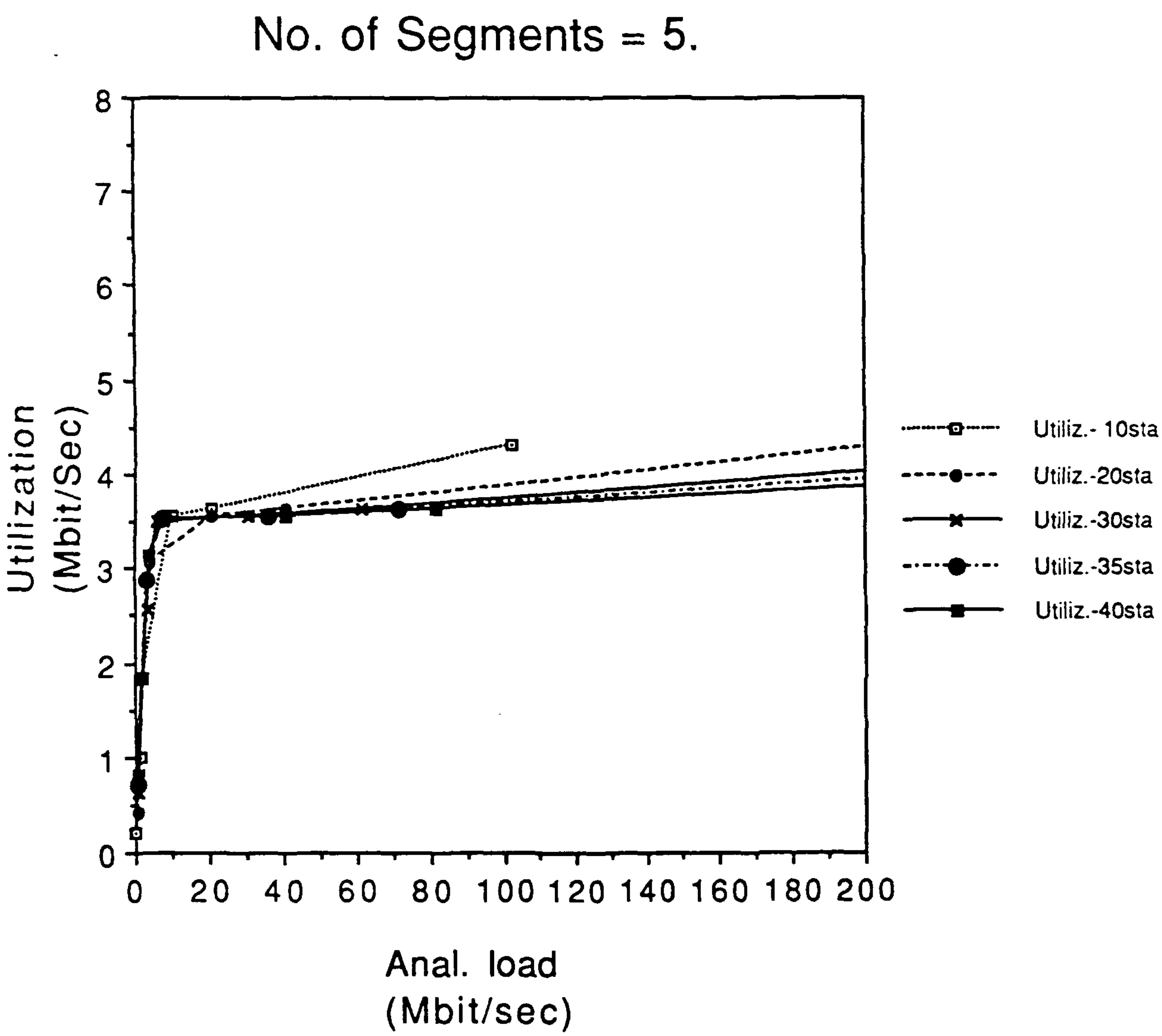
Graph 23


No. of Segments = 3.
8
7
6
5
4
3
2
1
0
Utilization (Mbit/sec)
0 20 40 60 80 100 120 140 160 180 200
Anal. load (Mbit/sec)
Utiliz.- 10sta
Utiliz.-20sta
Utiliz.-30sta
Utiliz.-35sta
Utiliz.-40sta

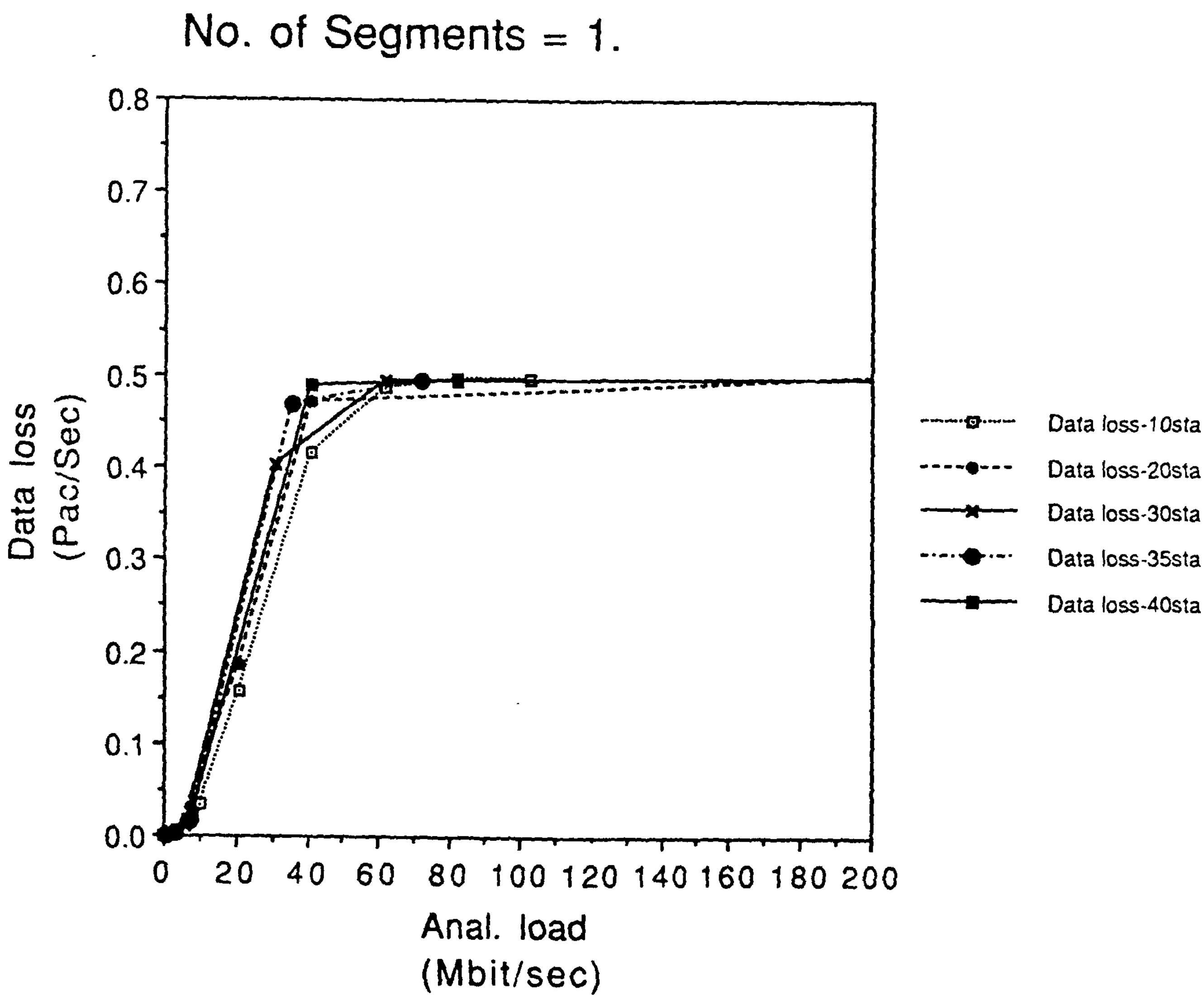
Graph 24


No. of Segments = 4.
Utilization (Mbit/Sec)
Anal. load (Mbit/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Utiliz.- 10sta
Utiliz.-20sta
Utiliz.-30sta
Utiliz.-35sta
Utiliz.-40sta

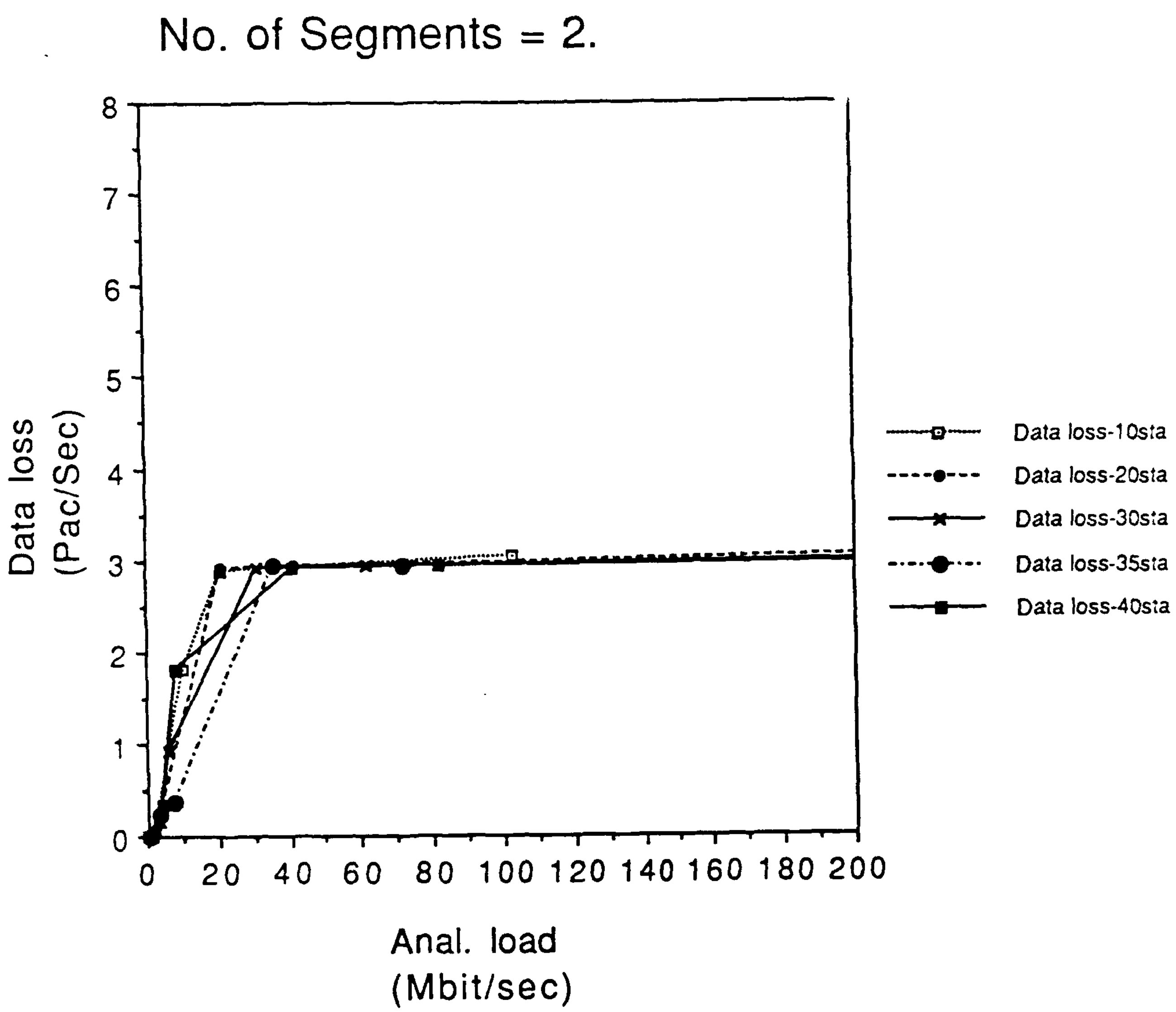
Graph 25


No. of Segments = 5.
Utilization
(Mbit/Sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbit/sec)
Utiliz.- 10sta
Utiliz.-20sta
Utiliz.-30sta
Utiliz.-35sta
Utiliz.-40sta

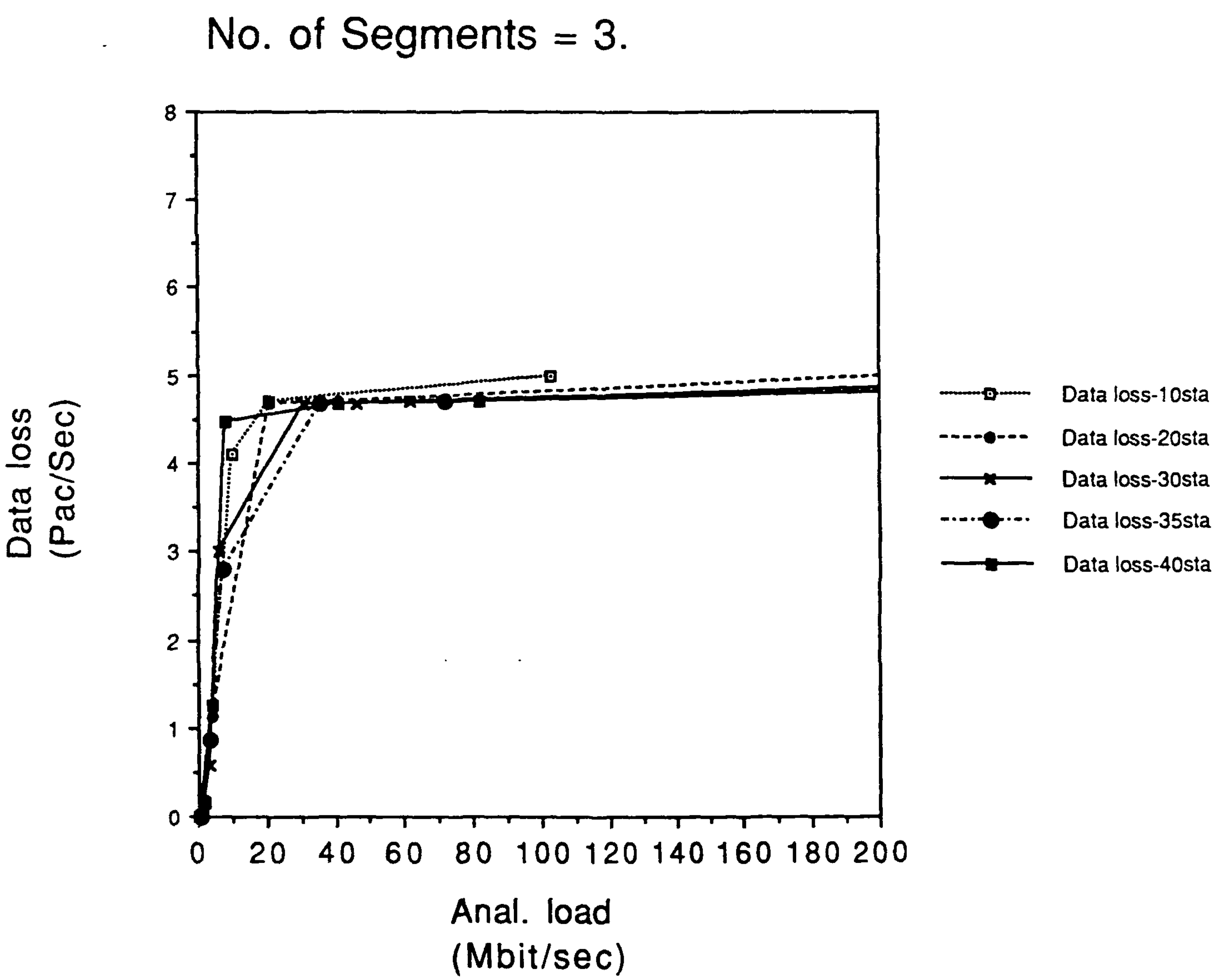
Graph 26


No. of Segments = 1.
Data loss
(Pac/Sec)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100
120
140
160
180
200
Anal. load
(Mbit/sec)
Data loss-10sta
Data loss-20sta
Data loss-30sta
Data loss-35sta
Data loss-40sta

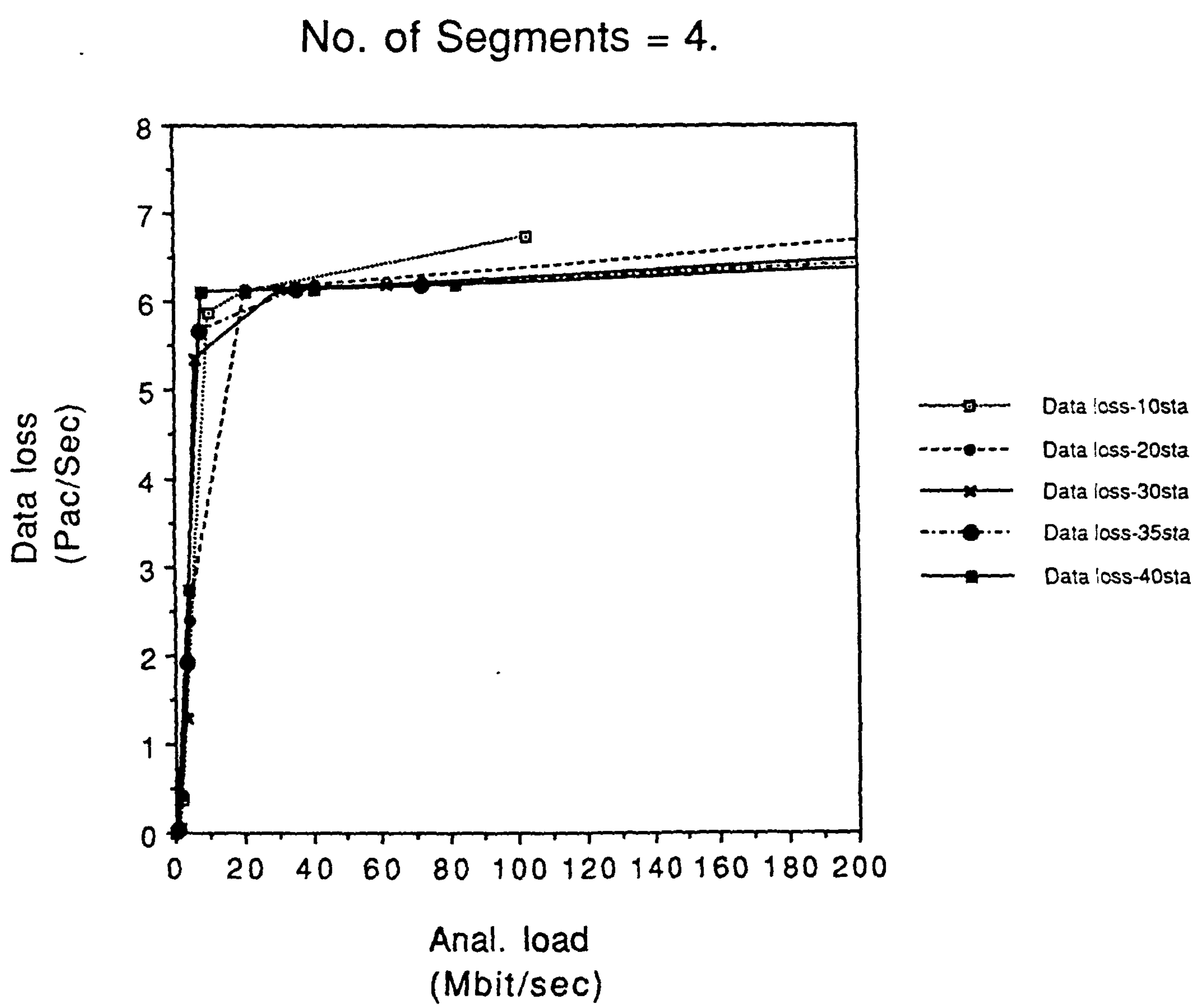
Graph 27


No. of Segments = 2.
Data loss (Pac/Sec)
Anal. load (Mbit/sec)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
180
200
Data loss-10sta
Data loss-20sta
Data loss-30sta
Data loss-35sta
Data loss-40sta

Graph 28

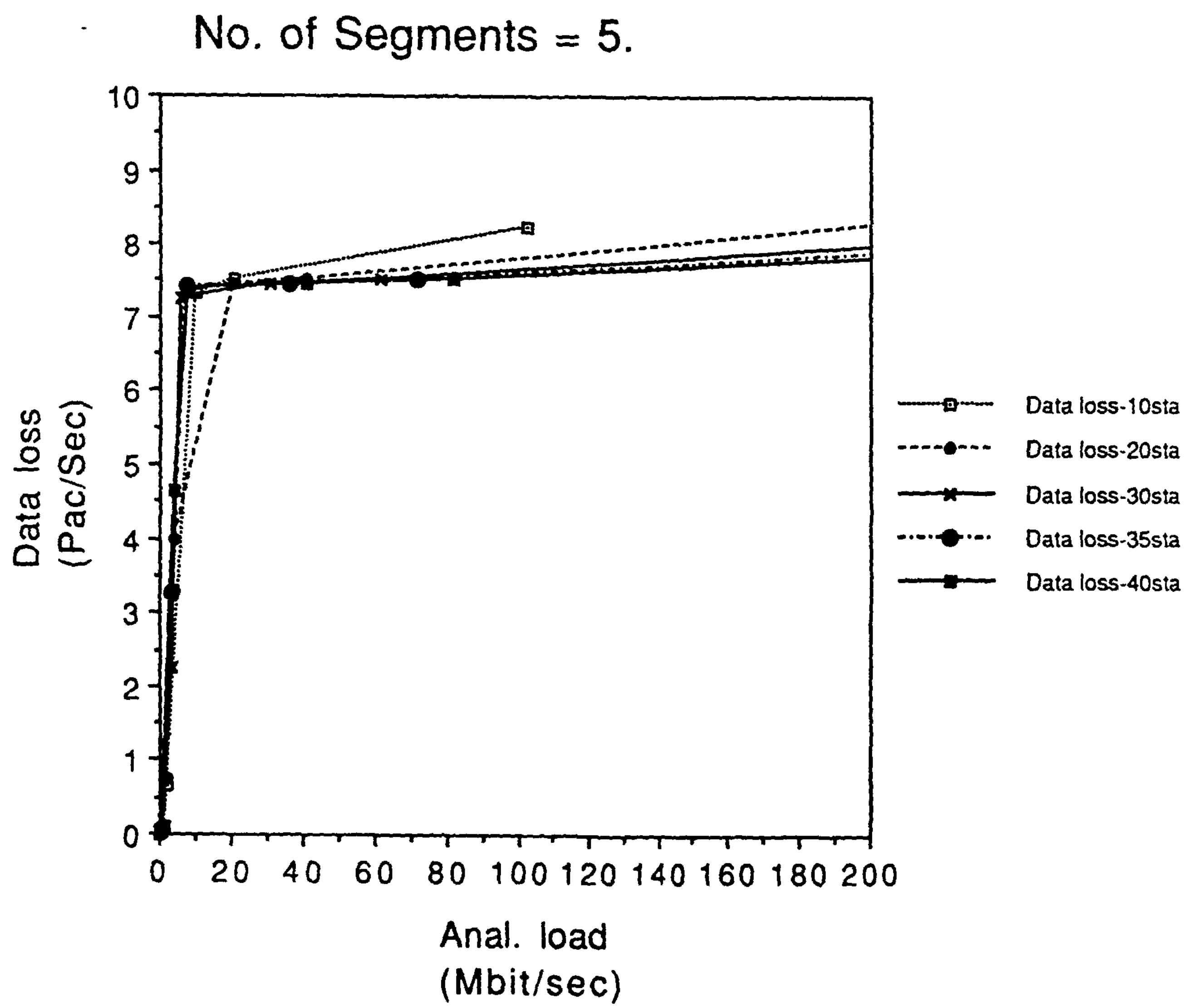

No. of Segments = 3.
Data loss (Pac/Sec)
0
1
2
3
4
5
6
7
8
0 20 40 60 80 100 120 140 160 180 200
Anal. load (Mbit/sec)
Data loss-10sta
Data loss-20sta
Data loss-30sta
Data loss-35sta
Data loss-40sta

Graph 29



No. of Segments = 4.
Data loss (Pac/Sec)
8
7
6
5
4
3
2
1
0
0
20
40
60
80
100
120
140
160
180
200
Anal. load (Mbit/sec)
Data loss-10sta
Data loss-20sta
Data loss-30sta
Data loss-35sta
Data loss-40sta

Graph 30



No. of Segments = 5.
10
9
8
7
6
5
4
3
2
1
0
Data loss
(Pac/Sec)
0 20 40 60 80 100 120 140 160 180 200
Anal. load
(Mbit/sec)
Data loss-10sta
Data loss-20sta
Data loss-30sta
Data loss-35sta
Data loss-40sta